

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESARROLLO DE UN SISTEMA DE ANÁLISIS DE
MOVILIDAD Y DETECCIÓN DE CAÍDAS
MEDIANTE APLICACIÓN ANDROID

GRADO EN INGENIERÍA DE
TECNOLOGÍA DE TELECOMUNICACIÓN

ALBERTO MILANÉS RODRÍGUEZ
MÁLAGA, 2021

Desarrollo de un sistema de análisis de movilidad y detección de caídas mediante aplicación Android

Autor: Alberto Milanés Rodríguez

Tutor: Eduardo Casilari Pérez

Departamento: Departamento de Tecnología Electrónica

Titulación: Grado en Ingeniería de Tecnologías de Telecomunicación

Palabras clave: detección de caídas, captura de movimiento, acelerómetro, giróscopo, Android Studio, Firebase, Kotlin.

Resumen

El proyecto desarrollado consiste en una aplicación implementada para dispositivos Android, al ser el sistema operativo dominante en el mercado. La *app* ofrece dos funcionalidades. Por un lado, permite la detección de caídas, servicio con la capacidad de dar avisos a un contacto de emergencia para ayudar a socorrer al usuario accidentado. Por otro lado, la aplicación posee un sistema de lectura de movimiento a través de los sensores del dispositivo móvil, lo que permitirá el desarrollo y creación de bases de datos para la configuración y evaluación de sistemas de seguimiento de actividad con mejores prestaciones. El proyecto se apoya sobre la plataforma Firebase, empleada para la gestión de usuarios y como base de datos, tanto para los datos de perfil como para los ficheros creados con la información sobre las actividades realizadas.

Development of fall detection and mobility analysis systems using an Android application

Author: Alberto Milanés Rodríguez

Supervisor: Eduardo Casilari Pérez

Department: Departamento de Tecnología Electrónica

Degree: Grado en Ingeniería de Tecnologías de Telecomunicación

Keywords: fall detection, motion capture, accelerometer, gyroscope, Android Studio, Firebase, Kotlin.

Abstract

The developed project consists of an application implemented for Android devices, as it is the dominant operating system on the market. The app has two functionalities. On the one hand, it operates as a fall detection, a service with the ability to alert an emergency contact to assist a fallen user. On the other hand, the application has a movement reading system through the sensors of the mobile device. This functionality enables the creation of movement datasets aimed at evaluating and developing human activity recognition and monitoring systems. The entire project is built on the Firebase platform, used for user management and as a database, both for the profile data and for the files created with the information on the monitored activities.

Agradecimientos

Quiero dar mi más sincero agradecimiento a mi familia y amigos por todo el apoyo mostrado durante todo este tiempo, a Marina por los ánimos que me ha dado cada día y a todos los profesores que me han formado este tiempo, en especial a Eduardo Casilari por su gran trabajo y dedicación como profesor. Ha sido un camino largo y duro que al fin termina.

*Dedicado a Marina,
a mi familia
y a la memoria de mi abuelo*

Contenido

Capítulo 1. Introducción	1
1.1. Contexto tecnológico	3
1.1.1. Reloj Localizador GPS B6	4
1.1.2. Reloj GPS S4 Mayores	4
1.1.3. Detector de caídas GPS Nook Senior	5
1.1.4. M-GUARD PRO	5
1.1.5. Apple Watch	6
1.2. Objetivos del Trabajo Fin de Grado	6
1.3. Estructura de la memoria	7
Capítulo 2. Especificaciones del sistema	9
2.1. Requisitos	10
2.2. Análisis y elección de tecnologías.....	12
Capítulo 3. Desarrollo del sistema	17
3.1. Entorno de desarrollo	17
3.2. Desarrollo software	19
3.2.1. Flujograma general.....	20
3.2.2. Monitorización de la actividad	23
3.2.3. Monitorización de caídas	25
3.2.4. Diagrama de clases	27
3.3. Estructura de la base de datos.....	32
Capítulo 4. Verificación y pruebas.....	35
4.1. Sistema de pruebas	35
4.2. Pruebas realizadas	37
Capítulo 5. Base de datos de movimientos	41

5.1. Organización de la base de datos.....	41
5.2. Experimento de captura de movimientos	42
Capítulo 6. Manual de instalación y uso	45
6.1. Manual de instalación	45
6.2. Manual de uso	48
6.2.1. Primera ejecución de la aplicación	48
6.2.2. Creación de un nuevo usuario	50
6.2.3. Inicio de sesión con un usuario creado	52
6.2.4. Menú principal	53
6.2.5. Configuración de la aplicación.....	55
6.2.6. Monitorización de actividad	56
6.2.7. Monitorización de caídas	59
6.2.8. Cierre de sesión del usuario.	62
Capítulo 7. Conclusiones y trabajo futuro.....	65
7.1. Tareas realizadas	65
7.2. Discusión y conclusiones	66
7.3. Líneas futuras.....	67
Apéndice A. Presupuesto de elaboración	69
Apéndice B. Estructura del DVD	71

Lista de Acrónimos

ADB	Android Debug Brigde
APK	Android aPplication Package
CSV	Comma Separated Value
GPS	Global Positioning System
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
INE	Instituto Nacional de Estadística
SDK	Software Development Kit
SMS	Short Message System
SMV	Signal Magnitude Vector
USB	Universal Serial Bus

Capítulo 1. Introducción

Una de las consecuencias más significativas de los avances de la tecnología y la medicina ha sido incrementar la esperanza de vida de las personas, lo que ha lugar a que el número de las denominadas “personas mayores”, consideradas como aquellas personas con 65 años o más, se incremente cada año.

Mediante la información disponible en el INE, Instituto Nacional de Estadística, puede observarse la distribución demográfica del país [1], centrándonos en aquellos mayores de 65 años. Mediante un cribado de información, puede verse en la Figura 1.1 como se distribuyen las personas mayores en el territorio nacional, habiendo una mayor concentración en la zona norte del país y siendo el valor promedio de la población mayor de 65 años el 19,77%.

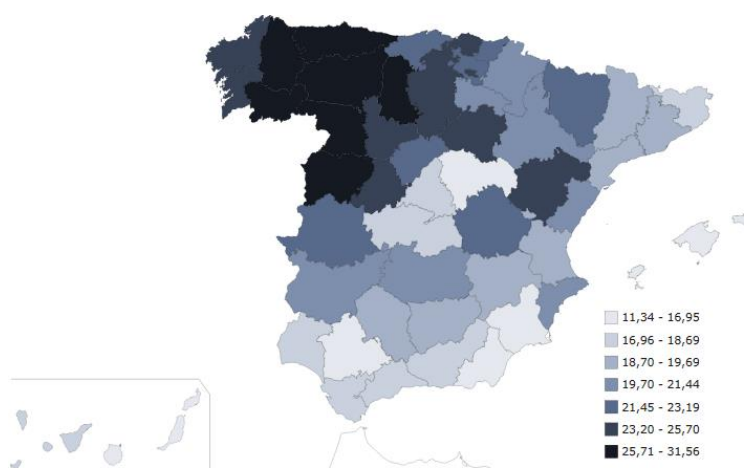


Figura 1.1. Porcentaje de personas mayores de 65 años por provincia. Fuente [1].

Consultando el censo del 1 de enero del 2021 [2] disponible en el propio INE la población española está cifrada en aproximadamente 47,4 millones de ciudadanos, de los cuales, las personas mayores suponen aproximadamente 9,4 millones.

Uno de los factores previamente mencionados responsable del incremento de personas mayores en la sociedad actual ha sido una mayor esperanza de vida. A través de los datos del INE puede accederse a una proyección del aumento de la esperanza de vida de hombres y mujeres [3], desde el 2020 hasta el 2069, que indica un incremento de 5 años para ambos sexos, como puede verse en la Figura 1.2.

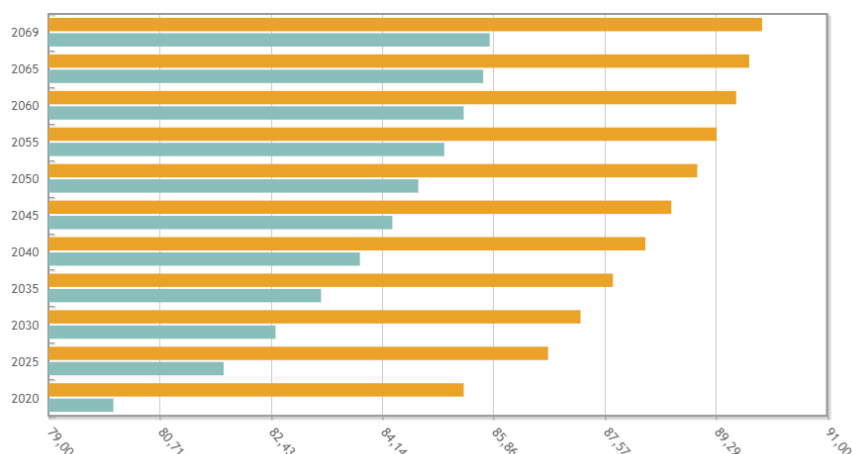


Figura 1.2. Proyeccion de la esperanza de vida. Fuente [3].

Otro aspecto para tener en consideración es el número de viviendas unipersonales habitadas por personas mayores [4], información brindada también por el INE en la que, una vez filtrados los datos de interés, se obtiene la distribución por comunidades autónomas mostrada en la Figura 1.3. En esta figura la información se muestra en miles de hogares y se observa que Andalucía y Cataluña son las que mayor concentración de viviendas con ancianos que viven solos presentan.

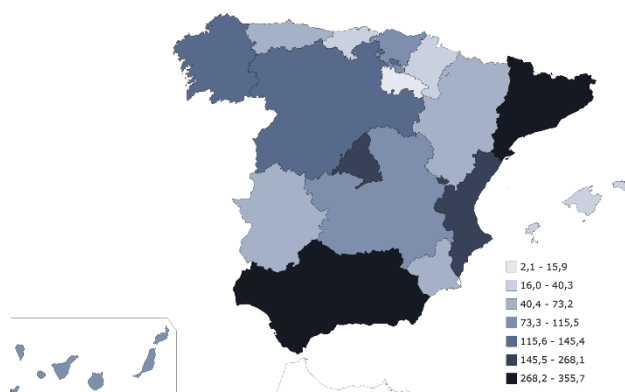


Figura 1.3. Hogares unipersonales por comunidad autónoma. Fuente [4].

Por otra parte, un artículo publicado en la Revista Española de Geriatria y Gerontología [5] en el que se realizó un seguimiento a 722 ancianos españoles durante un año, reveló que el 28,4% de los participantes sufrió una o más caídas anuales, de los cuales, el 9,3% sufrió fracturas y el 55,4% requirió de asistencia médica. Todos estos datos demuestran la existencia de un grave problema de salud pública vinculado a la cantidad de ancianos.

Si un accedente involucra un golpe en la cabeza, existe la posibilidad de que la persona quede inconsciente y con la posibilidad de algún tipo de lesión que, de no ser atendida con urgencia, puede ocasionar problemas de salud serios e incluso causar la muerte de la persona. Aunque no haya pérdida de conocimiento, es común que la persona accidentada sea incapaz de levantarse por sus propios medios para pedir ayuda, lo que puede dar lugar a síntomas de hipotermia y deshidratación, que pueden acabar conduciendo al mismo trágico final. Resulta de vital importancia tener la capacidad de responder a este tipo de sucesos con la mayor rapidez posible, para así poder salvar un número mayor de vidas.

1.1. Contexto tecnológico

La tecnología actual ha llegado a un punto de miniaturización capaz de integrar en un chip componentes de tamaño considerable como lo eran el acelerómetro, el giróscopo y el magnetómetro, todos ellos integrados en lo que conoce como sensor IMU, *Inertial Measurement Unit* o Unidad de Medida Inercial.

Esta capacidad de integración junto a la popularidad de los *smartphones* y los dispositivos conocidos como *wearables* o “vestibles” ha dado lugar a numerosas empresas enfocadas al diseño de este tipo de dispositivos, enfocados a la detección de caídas, tanto de personas mayores como deportistas y trabajadores de diversos campos.

A continuación, se presentan una serie de dispositivos comerciales con características similares a la aplicación que se busca desarrollar, así como su precio, con el fin justificar la necesidad de una aplicación de bajo coste para el usuario.

1.1.1. Reloj Localizador GPS B6

Es un modelo ofrecido por la empresa TRACmi [6] cuyo diseño se muestra en la Figura 1.4. Entre sus características más destacadas están presentes la detección fuera del domicilio y recordatorios de citas y medicación.



Figura 1.4. Reloj localizador GPS B6. Fuente [6].

Hay que tener en cuenta que este dispositivo carece de la funcionalidad principal, la detección de caídas. Su precio es de 99 € a que se añade una suscripción mensual de 19 €.

1.1.2. Reloj GPS S4 Mayores

Se trata del segundo tipo de dispositivos ofrecidos por la empresa TRACmi [7], mostrado en la Figura 1.5. Entre sus características principales está la localización fuera del domicilio y el establecimiento de una zona segura y la detección de caídas por control de inactividad.



Figura 1.5. Reloj GPS S4 Mayores. Fuente [7].

El principal problema se encuentra en que el fabricante no especifica el funcionamiento de la detección por control de inactividad, dando a entender que

solo se da el aviso en caso de que el usuario no sea capaz de levantarse por sí mismo o quede inconsciente. Resulta problemático, pues alguien capaz de levantarse podría requerir atención sanitaria y el contacto de emergencia ignoraría este evento.

Es un dispositivo con un coste de 199 € con el requisito de una suscripción mensual de 19 €.

1.1.3. Detector de caídas GPS Nook Senior

Este dispositivo esta ofertado por la empresa NEKI [8], cuyo diseño se observa en la Figura 1.6. Se trata de la opción más completa en cuanto a prestaciones, pues ofrece geolocalización, establecimiento de zona de seguridad y detección de caídas al momento en que se producen. Su coste es de 139 € con una suscripción de 12 € al mes.



Figura 1.6. Detector de caídas GPS Nook Senior. Fuente [8].

1.1.4. M-GUARD PRO

Este modelo lo oferta la empresa PANION [9], cuyo diseño se presenta en la Figura 1.7. Entre sus características destacan la detección de caídas, la localización mediante GPS y una limitada funcionalidad como teléfono móvil. Tiene un coste de 199 € con el inconveniente de que el usuario debe gestionar la compra de una tarjeta SIM.



Figura 1.7. M-GUARD PRO. Fuente [9].

1.1.5. Apple Watch

Se trata de la gama de relojes inteligentes ofrecida por Apple [10], cuyo diseño se puede ver en la Figura 1.8. Son dispositivos con la capacidad de detectar caídas, avisando al usuario con una vibración, una alarma y una notificación. También es capaz de identificar el movimiento del usuario tras el impacto contra el suelo, para esperar a que el usuario confirme el aviso o mandarlo automáticamente tras un minuto de inactividad. A los contactos de emergencia da información sobre la geolocalización y la magnitud de la caída. Su precio es de 620 € añadiendo el coste de la correspondiente tarifa de telefonía.



Figura 1.8. Apple Watch. Fuente [10].

1.2. Objetivos del Trabajo Fin de Grado

El objetivo de este proyecto es desarrollar un sistema vestible basado en una aplicación de *smartphone* desarrollada para el sistema operativo Android que persigue dos objetivos principales.

El primero es crear un sistema de monitorización de actividad que permita crear una base de datos con las medidas del acelerómetro y giróscopo. Estos datos podrán ser utilizados para establecer patrones de movimiento que puedan usarse para entrenar inteligencias artificiales para desarrollar sistemas de detección de caídas más precisos.

El segundo objetivo es detectar la caída del usuario y notificar el accidente a través de diversos medios y con la información relevante al contacto de emergencia designado.

Para llevar a cabo este proyecto, el primer paso será un estudio de la difusión y uso de *smartphones* en la población española, además de una comparativa de uso de los diferentes sistemas operativos del mercado, permitiendo justificar el desarrollo para un sistema en específico.

Para poder maximizar la efectividad del sistema, se busca establecer un umbral personalizable que permita discriminar el evento de la caída de las actividades cotidianas del usuario, así como poder cancelar un aviso en caso de que se produzcan falsas alarmas.

1.3. Estructura de la memoria

Una vez establecido el marco tecnológico en que se desarrolla el proyecto y los objetivos perseguidos, a continuación, se presenta un desglose de los capítulos en los que se divide este documento.

El segundo capítulo presenta una serie de requisitos que deben cumplirse en el desarrollo de la aplicación, junto a una justificación de la tecnología elegida y el sistema operativo sobre el que se ejecutará.

El tercer capítulo se centra en el proceso de desarrollo de la aplicación, teniendo en cuenta las especificaciones impuestas.

Tras finalizar el proceso de desarrollo, se pasa a un proceso de verificación y pruebas, en el que se centra el cuarto capítulo. Las pruebas se realizarán en base a una serie de ejercicios predeterminados que buscan simular actividades cotidianas con el fin de establecer unos umbrales recomendados que permitan evitar falsas alarmas.

Esta memoria presenta un quinto capítulo que trata de la fase de experimentación con diferentes sujetos con los que crear una base de datos inicial empleando el sistema desarrollado.

El sexto capítulo presenta el manual de instalación y uso, en el que se explica en profundidad cómo instalar correctamente el APK, *Android aPplication Package*, obtenido tras compilación del proyecto y el proceso de configuración y uso para maximizar la funcionalidad del usuario.

El séptimo y último capítulo muestra las conclusiones obtenidas tras todo el proceso de desarrollo y el planteamiento de posibles líneas de futuro desarrollo de este proyecto.

Capítulo 2. Especificaciones del sistema

Se desea desarrollar una aplicación móvil para *smartphone* basada en el sistema operativo Android. La app contará con dos funciones básicas. Por un lado, la detección de caídas y envío de alarmas y, por otro, la monitorización de actividades mediante los sensores inerciales del teléfono con los que se creará una base de datos de movimientos orientados a la evaluación de sistemas de reconocimiento de actividades. En la Figura 2.1 se presenta un esquema simplificado de los agentes implicados en el sistema.

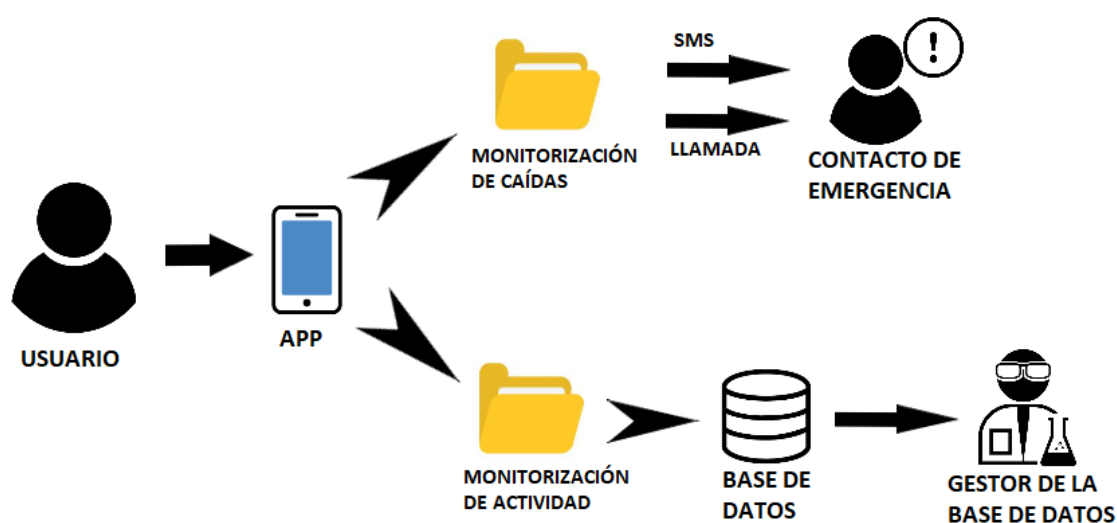


Figura 2.1. Agentes implicados en el sistema.

La aplicación cumple el rol de ser el agente principal en torno al cual se desarrollan dos grupos de agentes. El primer grupo de agentes está formado por el módulo de monitorización de caídas, el de monitorización de actividad y el sistema de almacenamiento. El segundo grupo de agentes se compone por el usuario, el contacto de emergencia y el gestor de la base de datos, que interactúan con el dispositivo y la información obtenida.

Una vez identificados los agentes, se establecen los siguientes casos de uso:

1. La persona que hace uso de la aplicación será capaz de identificarse de forma unívoca mediante la creación de un nombre de usuario y una contraseña, pudiendo introducir de forma opcional su nombre y el número del contacto de emergencia, junto a su edad, peso, altura y género.
2. El usuario podrá modificar la sensibilidad de la detección de caídas y la tasa de muestreo de los sensores, además de modificar o terminar de cumplimentar la información personal.
3. Mediante el botón correspondiente a la monitorización de caídas, se accede a una nueva ventana que permite iniciar y detener el proceso de seguimiento, bloqueando tanto el botón “atrás” del dispositivo como el botón “volver” de la aplicación, una vez se ha iniciado, hasta que se detenga la monitorización. Cuando se detecta una caída, se activa un aviso sonoro y aparece una ventana emergente con una cuenta atrás que permite cancelar el aviso, o notificar el accidente de forma manual o una vez ha expirado el tiempo.
4. A través del botón dedicado a la monitorización de actividad, la aplicación abre una nueva ventana que permite seleccionar varias actividades predefinidas o crear ejercicios personalizados, exigiendo en este último caso el nombre de dicha actividad. Una vez iniciada la actividad, no puede volverse a la ventana anterior hasta detener la grabación, momento en que se activa una barra de progreso que indica la carga del fichero en el servidor.

2.1. Requisitos

Para establecer los requerimientos del software necesarios para que el sistema funcione correctamente, se van a definir una serie de requisitos en la Tabla 2.1. Estas restricciones técnicas se clasifican en requisitos de interfaz, de monitorización de caídas y de monitorización de actividad.

Id.	Bloque funcional	Descripción	Tipo
1	General	Emplear el sistema operativo con mayor difusión en el mercado.	Restricción técnica

Tabla 2.1 Requisitos del sistema

Id.	Bloque funcional	Descripción	Tipo
2	General	La versión del software no puede estar obsoleta y debe ser compatible con la mayoría de los dispositivos a fecha de junio de 2021.	Restricción técnica
3	Interfaz	El sistema solicitará información de acceso para crear un nuevo usuario.	Funcional
4	Interfaz	El usuario podrá añadir información personal como su nombre, edad, peso, altura, género y número de contacto de emergencia.	Funcional
5	Interfaz	La información del usuario se almacenará en el dispositivo y en el servidor.	Funcional
6	Interfaz	El sistema permitirá iniciar sesión con un usuario ya creado.	Funcional
7	Interfaz	El usuario podrá modificar la información personal introducida.	Funcional
8	Interfaz	Al cerrar sesión, el sistema eliminará la información personal del dispositivo.	No funcional
9	Interfaz	El usuario podrá modificar la tasa de muestreo de los sensores, seleccionando uno de las posibles opciones ofrecidas por el sistema operativo.	Funcional
10	Interfaz	El usuario podrá modificar los umbrales para la sensibilidad de la caída y la sensibilidad de impacto para el sistema detector de caídas.	Funcional
11	Monitor de caídas	El sistema utilizará el acelerómetro disponible en el <i>smartphone</i> para monitorizar caídas.	Funcional
12	Monitor de actividad	El sistema utilizará el acelerómetro y el giróscopo integrados en el <i>smartphone</i> para monitorizar la actividad.	Funcional
13	Monitor de caídas	El sistema verificará que se ha introducido el nombre y el teléfono de contacto antes de acceder a la monitorización de caídas.	Funcional
14	Monitor de caídas	Tras detectarse la posible caída, el sistema lanzará un aviso acústico al usuario.	Funcional
15	Monitor de caídas	Tras detectarse una caída, el usuario podrá cancelar manualmente el aviso durante un minuto en caso de producirse una falsa alarma.	Funcional
16	Monitor de caídas	Tras terminar la cuenta atrás o dar la alarma manualmente, se establecerá una llamada con el contacto de emergencia.	Funcional
17	Monitor de caídas	La llamada realizada deberá tener activado el altavoz del dispositivo.	Restricción
18	Monitor de caídas	Tras expirar la cuenta atrás o dar el aviso manualmente, se enviará un SMS al contacto de emergencia.	Funcional
19	Monitor de caídas	En caso de que haya señal GPS, el SMS contendrá una URL con la geolocalización del usuario.	Funcional
20	Monitor de caídas	Si la señal es débil o el GPS está desactivado, el SMS avisará del accidente sin incluir información GPS.	Funcional
21	Monitor de caídas	La monitorización de caídas la iniciará el usuario mediante un botón en la app y bloqueará el regreso a la ventana anterior hasta que se detenga el seguimiento.	Funcional
22	Interfaz	El inicio y detención del seguimiento se avisará mediante mensajes en la interfaz.	No funcional

Tabla 2.1 Requisitos del sistema (Continuación)

Id.	Bloque funcional	Descripción	Tipo
23	Interfaz	El sistema será capaz de comprobar que se dispone de la información sobre edad, altura, peso y género antes de acceder a la monitorización de actividad.	Funcional
24	Interfaz	El usuario dispondrá de una serie de actividades predefinidas.	Funcional
25	Interfaz	Además de las actividades predefinidas, el usuario podrá registrar sus propias actividades, asignándoles un nombre, con el que podrá indicar qué acciones realizará durante la monitorización.	Funcional
26	Monitor de actividad	El sistema creará un fichero con formato csv.	Funcional
27	Monitor de actividad	El fichero contendrá una cabecera con la información del usuario, del dispositivo y la fecha de inicio de la actividad.	No funcional
28	Monitor de actividad	En el fichero se almacenará una marca de tiempo con las mediciones del acelerómetro y el giróscopo realizadas por el smartphone durante el periodo de monitorización.	No funcional
29	Monitor de actividad	Una vez detenido el registro de actividad, se almacenará el fichero en su correspondiente carpeta en el servidor, borrando el archivo original del dispositivo.	No funcional
30	Monitor de actividad	La monitorización de actividad se iniciará mediante un botón y bloqueará el regreso a la ventana anterior hasta que se detenga el seguimiento.	Funcional
31	Interfaz	El inicio y detención del registro se avisará mediante mensajes en la interfaz.	No funcional
32	Restricción técnica	Los procesos que hacen uso de los sensores deberán funcionar al mismo rendimiento, tanto con la pantalla encendida como con la pantalla apagada.	Restricción técnica

Tabla 2.1 Requisitos del sistema (Continuación)

2.2. Análisis y elección de tecnologías

Tras establecer los requisitos, se procede a la selección de tecnologías que se ajusten de la mejor forma posible al eje central del sistema, el *smartphone*. A continuación, se presentará un análisis tecnológico y comercial del que se extraerán los cuatro elementos principales:

- Sistema operativo del dispositivo.
- Método de almacenamiento de datos.
- Sistema de monitorización de actividad y almacenamiento de muestras de movilidad.
- Algoritmo de detección de caídas.

En el mercado han existido una amplia variedad de sistemas operativos para dispositivos móviles, siendo los más destacados Symbian, Blackberry, Windows Phone, IOS y Android. En la Figura 2.2 se presenta una distribución del peso en el mercado de los diferentes sistemas operativos desde el año 2009 al 2021 [11].

En el gráfico destaca como el sistema operativo Android paso del 0.66% en enero de 2009 a alcanzar el 72.44% en septiembre del presente 2021. Por otra parte, se observa cómo la presencia de Symbian fue decayendo hasta desaparecer en la actualidad y como Blackberry alcanzó una cierta popularidad en 2011 hasta desaparecer en la actualidad del mismo modo que Symbian. Por último, se debe considerar que la popularidad de IOS se ha mantenido relativamente constante hasta copar el 26.75% del mercado de dispositivos móviles en septiembre de 2021.

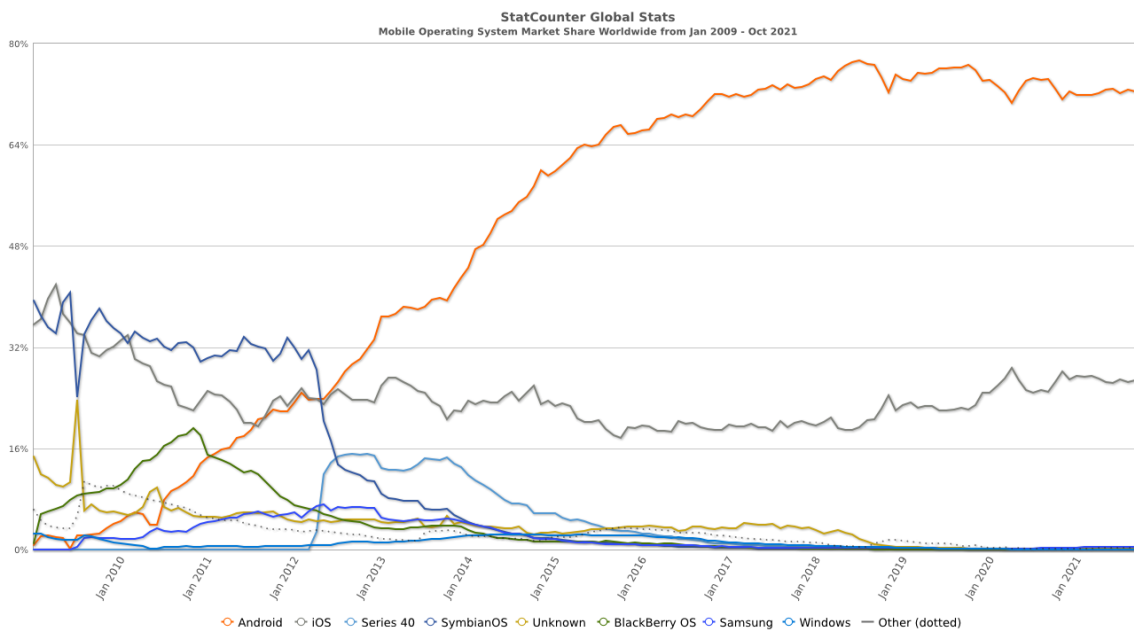


Figura 2.2. Cuota de mercado de sistemas operativos de dispositivos móviles. Fuente [11].

Debido a su amplia difusión en el mercado, para el desarrollo de este proyecto se empleará el sistema operativo Android. Se trata de una plataforma desarrollada por la empresa Google Inc., que ha recibido numerosas actualizaciones desde el momento de su creación. La Figura 2.3 muestra la distribución de versiones fechada en abril de 2020 [12].

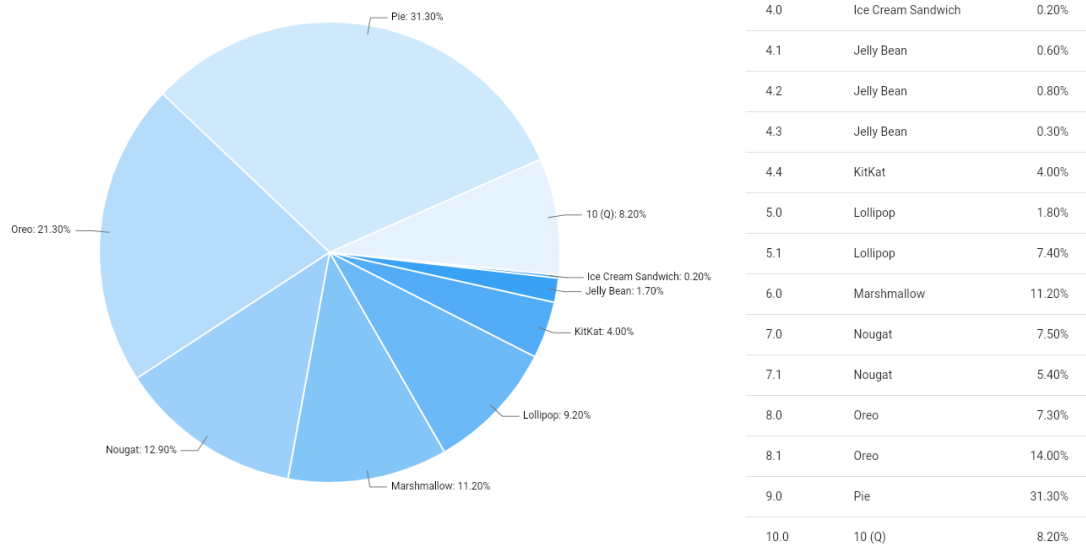


Figura 2.3. Distribucion de las versiones de Android. Fuente [12].

Para maximizar la compatibilidad con los dispositivos con el fin de alcanzar una mayor difusión, y teniendo en cuenta que las nuevas versiones de Android son compatibles con aplicaciones desarrolladas para versiones anteriores, se desarrollara la aplicación teniendo en cuenta la versión Marshmallow de Android, permitiendo que la aplicación se ejecute sin problemas en, aproximadamente, el 80% de dispositivos, sin tener que usar versiones obsoletas del software.

El segundo elemento en torno al cual se va a desarrollar el proyecto es el sistema de almacenamiento. Para implementar esta funcionalidad se ha escogido la plataforma Firebase, adquirida por Google en 2014 y que se integra en el marco de desarrollo de aplicaciones Android, lo que simplificará la gestión de la base datos al ofrecer una bibliografía dedicada para su implementación y la creación automática los recursos necesarios, asegurando la privacidad de los usuarios [13]. De todos los productos ofertados, se van a empelar los siguientes:

- *Firebase Auth*: es un servicio de autenticación de usuarios, pudiendo emplearse proveedores como Google o Facebook, además de los métodos clásicos como correo y contraseña.
- *Firebase Realtime Database*: se trata de un servicio de almacenamiento de información en tiempo real, mediante una base de datos alojada en la nube. Permite a los usuarios sincronizar y recuperar la información en cualquier dispositivo en que inicie sesión.

- *Firebase Cloud Storage*: este servicio permite el almacenamiento de archivos de cualquier tipo en la nube.

Para el proceso de monitorización de la actividad se tomarán mediciones del acelerómetro y el giróscopo integrados en el dispositivo con el objeto de generar patrones de movimiento para el entrenamiento de sistemas de inteligencia artificial orientada al reconocimiento de actividades. Los datos se almacenarán en ficheros con formato CSV, *Comma Separated Values* o Valores Separados por Comas. Para crear estos ficheros, hay que tener en cuenta la diferencia que existen entre las diferentes regiones a la hora de escribir números decimales, por lo que se ha optado por el estándar que emplean lenguajes de programación orientados al tratamiento de datos como Matlab o R.

El último aspecto a tratar es el algoritmo de detección de caídas. Se ha elegido un algoritmo básico que consiste en que, tras detectar la caída del módulo de la gravedad por debajo de un valor configurado por el usuario (lo que podría ser indicio de la conocida como “fase de caída libre” de una caída), si durante una ventana de un segundo se detecta asimismo un incremento de la aceleración superior a otro valor indicado por el usuario (posible indicio de un impacto contra el suelo), el sistema interpretará que se ha producido una caída.

Una vez establecido que el sistema operativo será Android, desarrollando la aplicación para la versión Marshmallow, empleando Firebase como método de almacenamiento de datos y los sistemas de monitorización de actividad y de caídas puede iniciarse el proceso de desarrollo del software que dará lugar al producto final y que se tratará en el siguiente capítulo.

Capítulo 3. Desarrollo del sistema

En este capítulo se va a presentar el entorno de desarrollo y el lenguaje de programación a partir de los cuales se va a implementar el software final de la aplicación. También se describirá el modelado software y los elementos más importantes que componen la arquitectura, finalizando con la descripción de la estructura de la base de datos empleada.

3.1. Entorno de desarrollo

Actualmente, en el mundo del desarrollo software existen multitud de IDE (*Integrated Development Environment*) con los que realizar los proyectos, desde códigos sencillos hasta complejas aplicaciones. A la hora de escoger un IDE para desarrollar un proyecto es importante considerar tanto los lenguajes de programación soportados como las facilidades que proporciona al desarrollador. Para este proyecto se ha escogido el lenguaje de programación Kotlin, lenguaje desarrollado por Google, ya que el producto final consiste en una aplicación Android.

Originalmente Android se podía desarrollar en diversas plataformas, aunque Google consideraba como opción oficial Eclipse [14], un IDE gratuito que soporta numerosos lenguajes y que ha sido adaptado a diversas necesidades en el campo de la ingeniería. Eclipse fue sustituido por Android Studio [15], un IDE de código abierto que presente mejoras respecto a Eclipse.

Para el desarrollo se ha seleccionado Android Studio por ser un entorno dedicado en exclusiva a implementar aplicaciones Android, con una gran comunidad de usuarios y que ofrece una serie de ventajas que favorecen y

facilitan el desarrollo a través del SDK (*Software Development Kit*). Entre las principales ventajas destacan:

- Uso de la herramienta *Gradle* para la gestión y automatización de la construcción de proyectos.
- Interfaz para desarrollar los *layouts* de forma intuitiva, combinada con el uso del metalenguaje XML.
- Alertas en tiempo real de errores sintácticos y de compatibilidad.
- Renderizado en tiempo real de los *layouts*.
- Ejecución de la ampliación mediante un emulador o mediante el dispositivo a través del puerto USB y la herramienta ADB, (*Android Debug Bridge*).
- Traducción de código en lenguaje Java a Kotlin y viceversa, permitiendo reciclar código de proyectos en otros lenguajes.

En la Figura 3.1 se muestra la organización de un proyecto en Android Studio, [16] con el fin de tener una visión más clara del proyecto.

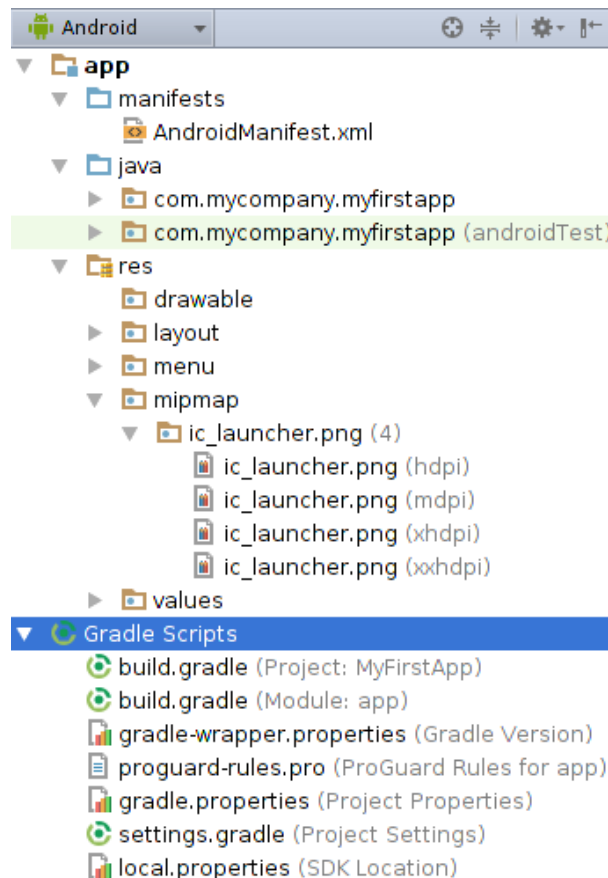


Figura 3.1. Estructura de un proyecto en Android Studio

3.2. Desarrollo software

Para desarrollar una aplicación Android, es necesario el uso de un lenguaje de programación, Java o Kotlin, habiéndose escogido el segundo, y una interfaz gráfica implementada mediante el metalenguaje XML.

La unión de la parte programática y la parte gráfica da lugar a una “actividad”, elemento que se comporta como una interfaz de usuario. Las actividades funcionan de forma independiente unas de otras, con mecanismos para transmitirse datos, y el conjunto de todas ellas originan una aplicación.

En la Figura 3.2 se muestra la arquitectura software de cualquier sistema Android [17]. La aplicación trabajará sobre la capa de aplicación haciendo uso de administradores del marco de aplicación con los que acceder a la información del acelerómetro, giróscopo y ubicación.

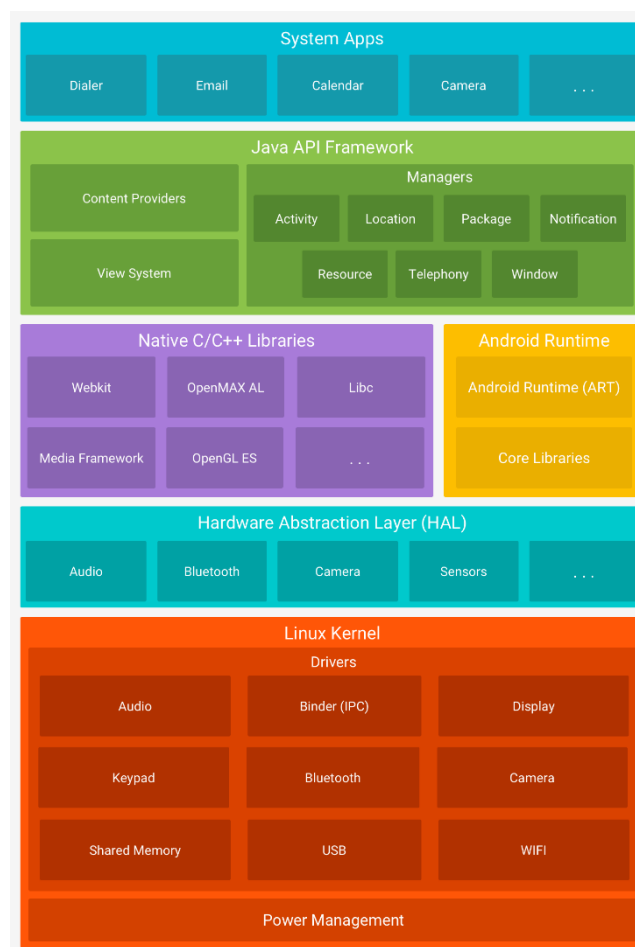


Figura 3.2. Arquitectura software Android

3.2.1. Flujograma general

Dada la complejidad de una aplicación Android y el número de actividades que la conforman, esta sección presenta un esquema del sistema desarrollado que facilite comprender su funcionamiento. El flujograma del sistema se va a dividir en dos partes, una primera en la que se presenta el proceso de autenticación (Figura 3.3) y una segunda en la que se muestra el funcionamiento general del resto del sistema (Figura 3.4).

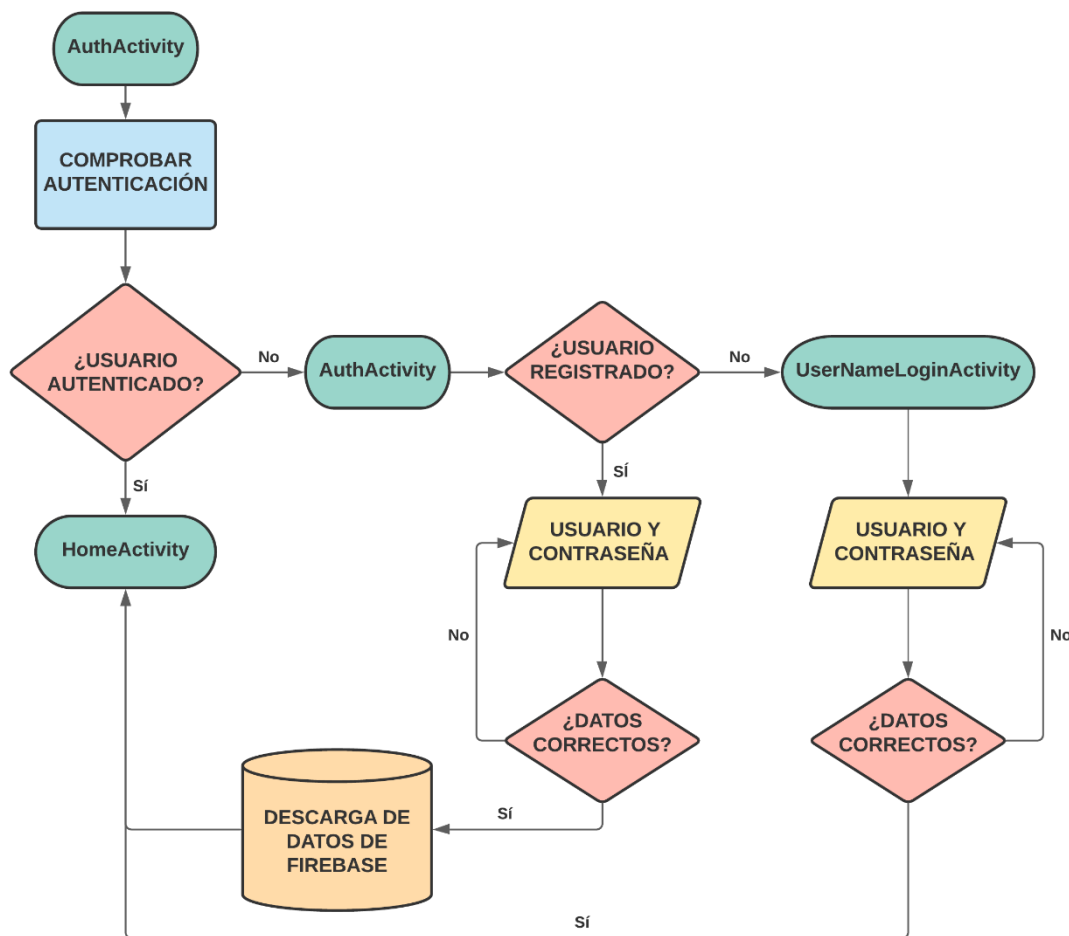


Figura 3.3. Flujograma del proceso de autenticación

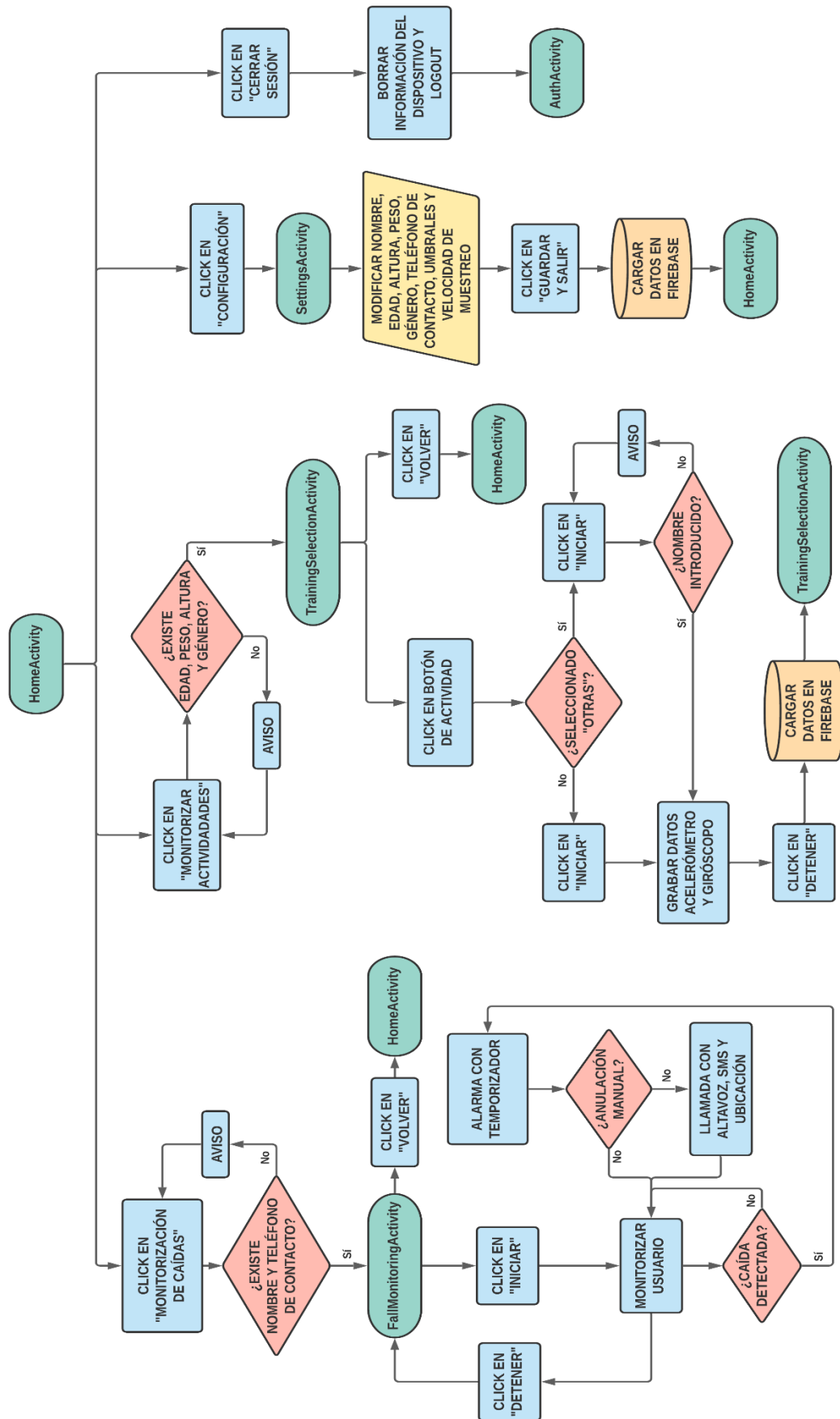


Figura 3.4. Flujograma del funcionamiento general de la aplicación

El flujograma de la Figura 3.3 representa el proceso de acceso al sistema. Una vez iniciada la aplicación, se comprueba si el usuario ya está autenticado, llevando directamente a la actividad *HomeActivity* o mostrando *AuthActivity* en caso contrario.

Si el usuario ya está registrado, podrá introducir su nombre de usuario y contraseña, abriendo *HomeActivity* y descargando la información personal del servidor, en caso de una autenticación exitosa. Si se produjo algún tipo de error se mostrará un mensaje.

En el caso de que el usuario no exista, podrá crearse, pulsando en “CREAR USUARIO”, lo que inicia la actividad *UserNameLoginActivity*. La información mínima que requiere la aplicación es el nombre de usuario y escribir dos veces la contraseña, y avisará de errores como que el usuario ya exista, la contraseña tenga algún error o en el caso de la introducción de caracteres no válidos. En este punto, también se puede añadir la información sobre edad, peso, altura, género y número de contacto.

Si la autenticación concluye satisfactoriamente, se cargan los datos en el servidor y se inicia *HomeActivity*, actividad desde la que inicia el flujograma de la Figura 3.4. En esta actividad se presentan los botones que dan acceso a las principales funcionalidades: monitorización de caídas y monitorización de actividad, un botón para acceder a la configuración y otro para el cierre de sesión.

Pulsando en el botón “MONITORIZACIÓN DE CAÍDAS” se comprueba que la aplicación dispone tanto del nombre del usuario como del teléfono de contacto de emergencia, siendo datos relevantes para la funcionalidad. En caso de que la comprobación sea errónea, da un aviso al usuario sobre la información requerida y marca en rojo el botón “CONFIGURACIÓN”. En el caso de que todo esté correcto, se inicia *FallMonitoringActivity*.

Cuando se inicia esta actividad, el seguimiento no comienza hasta que el usuario pulsa en “INICIAR”, bloqueando la posibilidad de volver a la actividad anterior hasta que pulsa en “DETENER”. En el momento en que se detecta una posible caída, se activa una notificación sonora junto a un mensaje con cuenta atrás en la aplicación que permite cancelar el aviso o dar la alarma de forma inmediata mediante un SMS con la ubicación y una llamada telefónica configurada para

tener activado el altavoz. Una vez termina la llamada, el proceso de monitorización de caídas permanece activo hasta que se cierre la aplicación o se pulse en el botón “DETENER”.

Con una pulsación en “MONITORIZAR ACTIVIDAD”, el software comprueba si el usuario ha introducido la información sobre su edad, peso, altura y género, pues es información importante para la base de datos. Si la comprobación fue correcta, se inicia *TrainingSelectionActivity* y en el caso contrario muestra un mensaje análogo al del caso anterior.

Esta actividad permite seleccionar una serie de ejercicios físicos, de los que obtener muestras mediante el acelerómetro y el giróscopo, abriendo *RecordDataActivity*. En caso de haber pulsado en “OTRAS”, esta actividad mostrara un cuadro de texto con el que introducir un nombre descriptivo con el que etiquetar dicho ejercicio. Para iniciar el seguimiento se pulsará en “INICIAR”, quedando bloqueado el volver a la actividad anterior hasta haber pulsado “DETENER”, momento en que el archivo con los datos capturados por los sensores se cargará en la base de datos de Firebase, que se describe en el apartado 3.3. Tras esa operación, la aplicación volverá al inicio del proceso *TrainingSelectionActivity*.

Si el usuario pulsa en “CONFIGURACIÓN”, accede a *SettingsActivity* donde puede modificar o terminar de añadir la información personal, modificar los umbrales para la detección de caídas y la tasa de muestreo de los sensores. Esta información se carga en el servidor, actualizando la configuración anterior del usuario.

Finalmente, tras pulsar en “CERRAR SESIÓN”, se cierra la sesión en Firebase y se borra la información introducida por el usuario del dispositivo, permaneciendo almacenada en el servidor para su descarga cuando el usuario vuelva a iniciar sesión en el mismo o en otro dispositivo, y devuelve la aplicación a *AuthActivity*.

3.2.2. Monitorización de la actividad

La forma en que se inicia la monitorización de la actividad ya se ha mencionado en el apartado anterior, por lo que este apartado tratará sobre el proceso de

adquisición de datos. Al iniciar el seguimiento de la actividad, se toman muestras del acelerómetro y giróscopo en un sistema de coordenadas tridimensional como muestra la Figura 3.5.

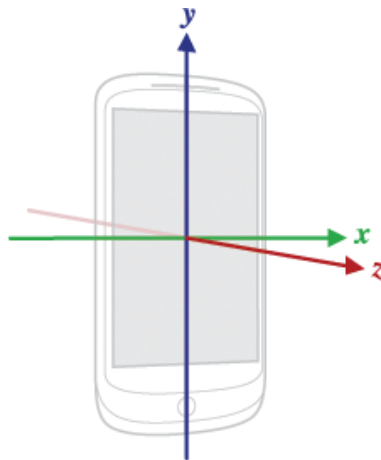


Figura 3.5. Sistema de coordenadas en relación al dispositivo

En los dispositivos Android los sensores asociados al acelerómetro y giróscopo presentan cuatro niveles de posibles velocidades de muestreo seleccionables por el usuario:

- `SENSOR_DELAY_FASTEST`
- `SENSOR_DELAY_GAME`
- `SENSOR_DELAY_UI`
- `SENSOR_DELAY_NORMAL`

Con el fin de facilitar la interacción del usuario con el sistema, se han seleccionado los tres primeros niveles de velocidad para poder asociarlos con botones mediante las etiquetas “ALTA”, “MEDIA” y “BAJA”.

Cuando se pulsa en “INICIAR”, se crea un fichero de tipo CSV (Comma Separated Values) con una cabecera con el nombre de usuario, sus datos personales, fecha y hora del inicio de la actividad, el tipo de ejercicio realizado, la tasa de muestreo, información del dispositivo y las unidades en las que se están midiendo los datos.

En el archivo CSV las medidas obtenidas se guardan en filas de siete valores, correspondiendo el primer valor con una marca de tiempo, las posiciones 2 a la 4 con las componentes x, y, z de la aceleración medida, mientras que las posiciones 5 a la 7 se reservan para medidas del giróscopo. Como Android no

permite obtener medidas simultaneas de aceleración y velocidad angular, cada fila solo representa las medidas de una de las dos magnitudes. La Figura 3.6 muestra un fragmento de uno de los ficheros.

	A	B	C	D	E	F	G
1	# User ID sujeto01						
2	# Peso: 107 -Altura: 190 -Edad: 29 -Genero: Masculino						
3	# Fecha y hora del comienzo de la prueba 27/10/2021 10:13:55						
4	# Tipo de actividad "Caminar"						
5	# Tipo de muestreo: SENSOR_DELAY_GAME						
6	# Características del dispositivo .Version ANDROID: 30 -Modelo: SM-A315G -Fabricante: samsung						
7	# Acelerometro [g] ---Giroscopo [rad/s]						
8	TimeStamp	ax	ay	az	gx	gy	gz
9	10:13:55.298	0.0600387	0.3458327	0.7548768			
10	10:13:55.301				-3.0659328	0.4263839	1.5430456
11	10:13:55.316	-0.2230706	0.281889	0.9567142			
12	10:13:55.319				-3.0659328	0.4263839	1.5430456
13	10:13:55.336	-0.4373551	0.3475411	1.21737			
14	10:13:55.339				-3.0659328	0.4263839	1.5430456
15	10:13:55.356	-0.3243554	0.2125761	1.4077366			
16	10:13:55.359				-3.0659328	0.4263839	1.5430456
17	10:13:55.376	-0.1256908	0.1737705	1.1744155			
18	10:13:55.379				-0.6181957	0.444099	0.1582141
19	10:13:55.396	0.0344124	0.065408	0.8546973			
20	10:13:55.399				-0.3475823	0.6939429	0.3915646
21	10:13:55.416	-0.0883496	0.2472325	0.9547618			
22	10:13:55.420				-0.8539896	0.8374963	0.3304781
23	10:13:55.436	-0.1745027	0.1684012	1.0421351			
24	10:13:55.440				-1.323745	0.580322	0.3115413
25	10:13:55.456	-0.1291076	0.0131792	1.1121803			
26	10:13:55.459				-1.2003502	0.7819075	0.4025602
27	10:13:55.476	-0.2679776	0.032704	1.172463			
28	10:13:55.479				-0.7232645	0.2962697	0.1875356
29	10:13:55.496	-0.2169691	0.1198333	1.0726426			

Figura 3.6. Fragmento inicial de un fichero CSV

Es importante tener en cuenta que, a pesar del nombre del formato, “valores separados por comas”, hay incompatibilidades con programas como Excel de Microsoft en el que estos valores se separan por punto y coma.

También hay que considerar el formato numérico de cada región, pues hay países en los que la parte decimal se separa mediante coma y otros en los que se usa el punto. Como los datos van a analizarse con programas de análisis numérico como Matlab, se va a emplear el punto y coma como separador de valores y el punto para indicar la parte decimal de los valores.

Una vez ha terminado la actividad, el fichero CSV se carga y almacena en la base de datos, en la carpeta correspondiente a la actividad realizada, y se elimina del dispositivo para liberar espacio del disco de almacenamiento del móvil.

3.2.3. Monitorización de caídas

Los requisitos para iniciar la actividad de monitorización de caídas ya han sido establecidos en el apartado anterior, por lo que esta sección se centrará en el algoritmo de detección de caídas.

La detección se lleva a cabo mediante un sencillo algoritmo representado en la Figura 3.7 en el que se analiza el módulo o SMV (*Signal Magnitude Vector*) del vector tridimensional obtenido del sensor acelerómetro.

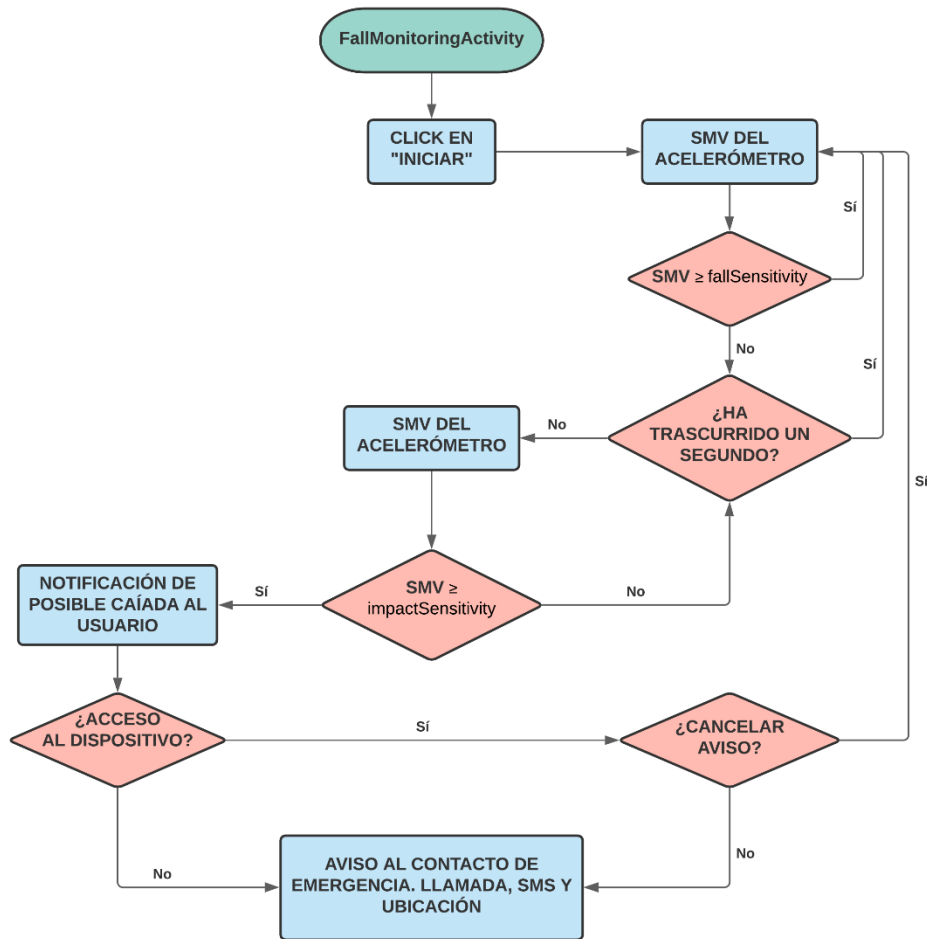


Figura 3.7. Flujograma del algoritmo de detección de caídas

Si el valor del SMV cae por debajo de un umbral configurado por el usuario, denominado *fallSensitivity*, se detecta una posible fase de caída libre en la que se desploma el módulo de la aceleración. En este momento se abre una ventana temporal de un segundo en el que, si el SMV supera un segundo umbral, también configurable por el usuario y denominado *impactSensitivity*, se interpreta como un indicio de impacto contra el suelo (suceso que normalmente provoca un pico brusco del valor de SMV).

Tras detectar la posible caída, el software notifica primero al usuario el evento, dando la posibilidad de cancelar el aviso si fue una falsa alarma. Si, tras el

impacto el usuario no es capaz de cancelar la alarma, esta se dispara tras una espera de un minuto.

El método a través del cual se transmite el aviso al contacto de emergencia se basa en un SMS con la ubicación del usuario y una llamada telefónica configurada para activar el altavoz, de forma que el accidentado o quien le esté prestando ayuda pueda hablar con el contacto de emergencia sin tener que coger el dispositivo.

3.2.4. Diagrama de clases

Este apartado se centrará en los diagramas de las clases más importantes que componen el conjunto del sistema, junto a los atributos y métodos más relevantes.

Todas las clases de la aplicación extienden la clase *AppCompatActivity*, además de algunas clases secundarias centradas en el uso de sensores y periféricos. En la Figura 3.8 se muestra esta relación.

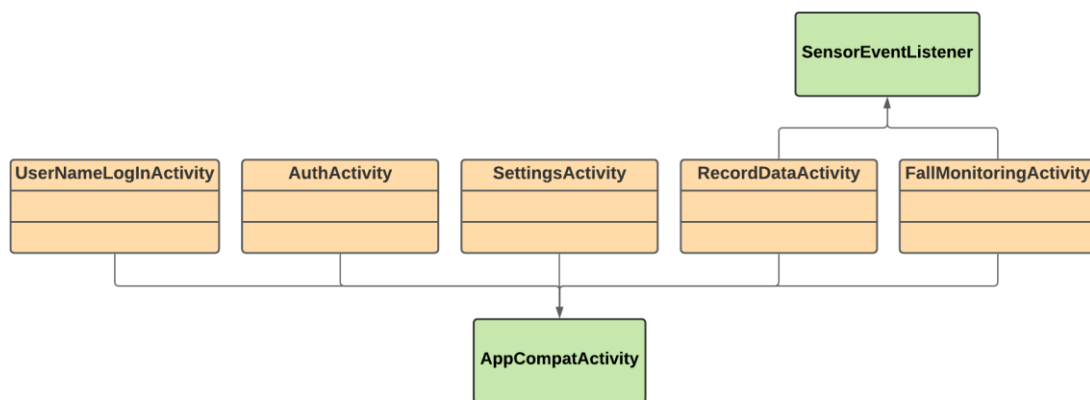


Figura 3.8. Relación entre las clases creadas y las asociadas

No se va a profundizar en las clases asociadas a las actividades pues su función es gestionar objetos del sistema. En cambio, análisis se centrará en las clases diseñadas para la aplicación.

En la Figura 3.9 se representa la clase *UserNameLoginActivity*, encargada de la creación de usuarios a través de los atributos *userName*, *password* y *repPassword*. Son los campos mínimos que exige la actividad para crear un

usuario, pudiendo asignar un valor a los atributos *name*, *age*, *height*, *weight*, *gender* y *mobile*.

Una de las funciones más importantes es *setup*, la cual se ocupa de la configuración inicial de la actividad. En segundo lugar, está *loginFunction* método encargado de crear el nuevo usuario en la plataforma Firebase tras la verificación de que los campos mínimos son correctos. Finalmente, está el método *saveUserData*, que se encarga de subir la información a la base de datos de Firebase.

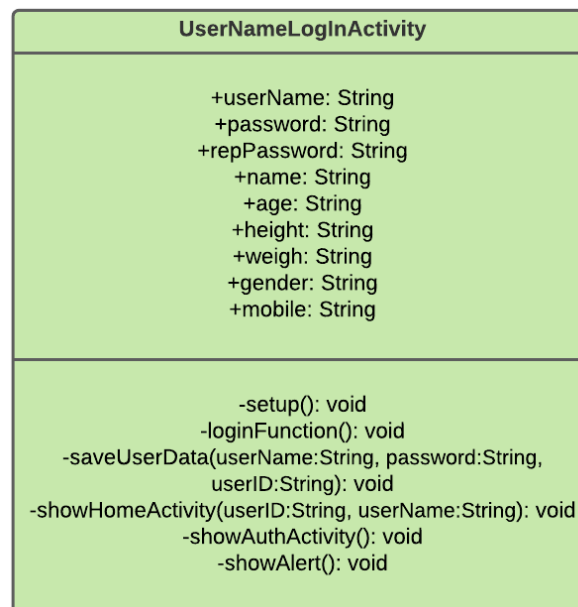


Figura 3.9. Diagrama de la clase *UserNameLoginActivity*

La clase *AuthActivity* se muestra en la Figura 3.10. Los atributos principales son *userName* y *password*, empujados por *userNameSignInFunction* para efectuar la autenticación en Firebase de un usuario ya existente.

Uno de los métodos fundamentales es *applicationPermissionRequestFunction*, empleado en la petición de permisos al usuario para disponer de acceso a las llamadas, ubicación, SMS y almacenamiento interno. Finalmente, se debe mencionar el método *loadSavedData* que se encarga de sincronizar en la aplicación la información de usuario cargada en la base de datos.

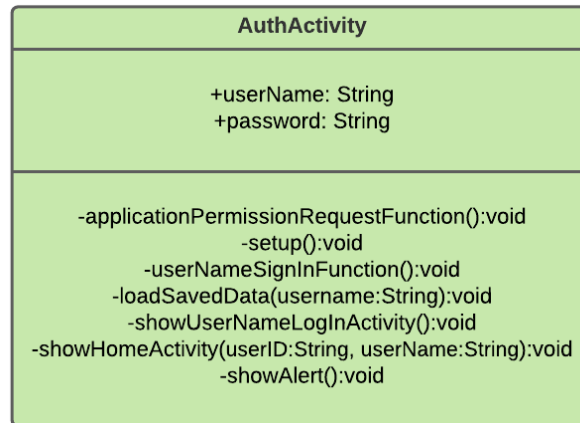


Figura 3.10. Diagrama de la clase *AuthActivity*

La siguiente clase de la que se va a hablar es *SettingsActivity*, mostrada en la Figura 3.11. Presenta atributos comunes con *UserNameLogInActivity*, ya que permite introducir dicha información si no se había añadido previamente o modificarla en caso de que sea necesario.

Por otro lado, la clase incluye los atributos *fallSensibilityBar* e *impactSensibilityBar*, que referencian los elementos de clase *SeekBar* de la interfaz de los que se obtiene los valores umbrales empleados por el algoritmo de detección de caídas. El primer atributo permite seleccionar un umbral comprendido entre 0.4g y 0.8g, mientras que el permite seleccionar valores en un rango desde 2g hasta 5g.

Por último, se presentan los atributos *uiToogleButton*, *gameToogleButton* y *fastestToogleButton* que referencian elementos de tipo *ToogleButton* de la interfaz. Se usan para que el usuario seleccione una tasa de muestreo (ALTA, MEDIA o BAJA).

De los diferentes métodos, *setup* realiza la misma función que en las clases anteriores, por su parte, *loadPreviousUserData* y *loadPreviousSettings* muestran en la interfaz los valores almacenados previamente en el dispositivo en tanto que *saveUserData* y *saveSettings* se ocupan de guardar los cambios tanto en el dispositivo como en el servidor.

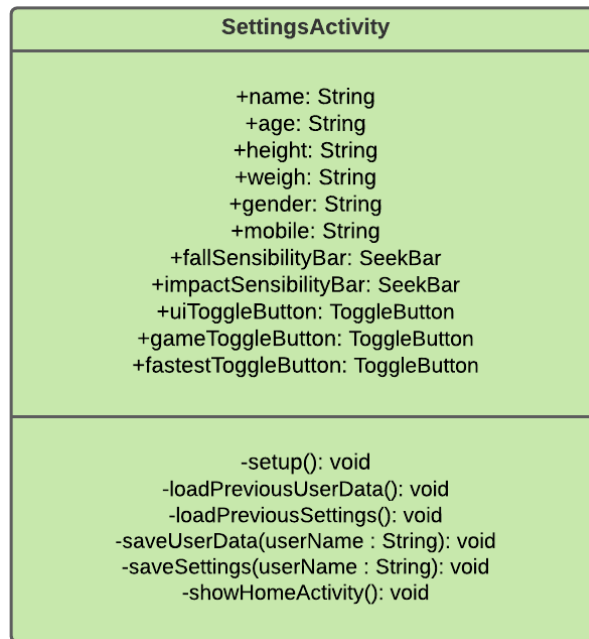


Figura 3.11. Diagrama de la clase *SettingsActivity*

La Figura 3.12 muestra la clase *RecordDataActivity*. Esta hace uso de los atributos vistos en clases anteriores para la creación de la cabecera del fichero CSV, referenciado por el atributo *fileOut* al que se le asigna el nombre almacenado en *fileName* que contiene información sobre el nombre de usuario, la actividad y la fecha y hora de inicio.

El último atributo importante es *sensorDelay* que se emplea en la configuración del acelerómetro y el giróscopo.

En relación a los métodos implementados, *setup* cumple la misma función que en las clases anteriores. El método *initTrainingRecordData* se ejecuta al pulsar el botón de inicio. Crea el fichero, con la cabecera y los identificadores de las columnas, terminando con el registro de periféricos y configurando la CPU para que, al apagar la pantalla, la aplicación funcione a pleno rendimiento.

Esta clase termina con el método *uploadTrainingRecordData* que se ejecuta al pulsar en el botón de detener, borrando los sensores del registro y devolviendo la CPU a su configuración por defecto. Termina cargando el fichero en la carpeta correspondiente del servidor de Firebase.

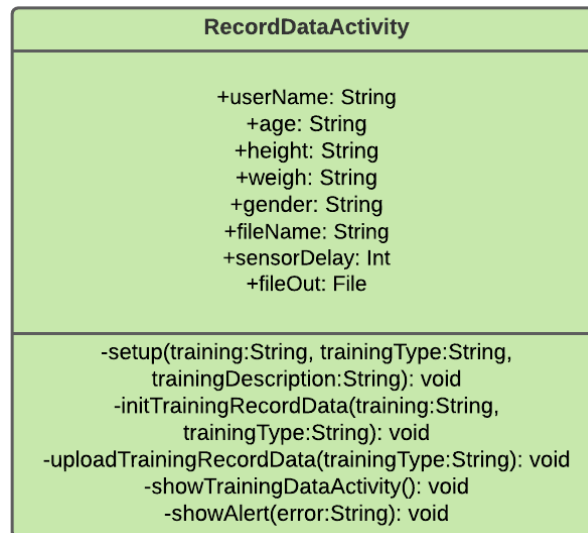


Figura 3.12. Diagrama de la clase *RecordDataActivity*

Para finalizar, se muestra en la Figura 3.13 la clase *FallMonitoringActivity*, encargada de la detección de caídas y el aviso al contacto de emergencia. Entre los atributos principales, destacan *fallSensitivity* e *impactSensitivity* que son los valores umbrales que configura el usuario para identificar la fase de caída libre y el impacto contra el suelo. Por otro lado, está el atributo *locationCallback* que referencia un método empleado en la geolocalización del usuario.

El método *processingData* contiene el algoritmo de detección de caídas. *notifyFall* y *createNotificationChannel* son los métodos encargados de avisar al usuario sobre la posible caída, para cancelar o dar el aviso de emergencia.

En cuanto a la ubicación, los métodos encargados de este propósito son *isLocationEnable*, *getNewLocation* y *getCurrentLocation*. La primera función verifica que el dispositivo tiene el GPS activado, mientras que las restantes se ocupan de obtener la ubicación del dispositivo.

Por último, los métodos *sendSMS* y *phoneCall* se ocupan de mandar el mensaje y de establecer la llamada con el contacto de emergencia.

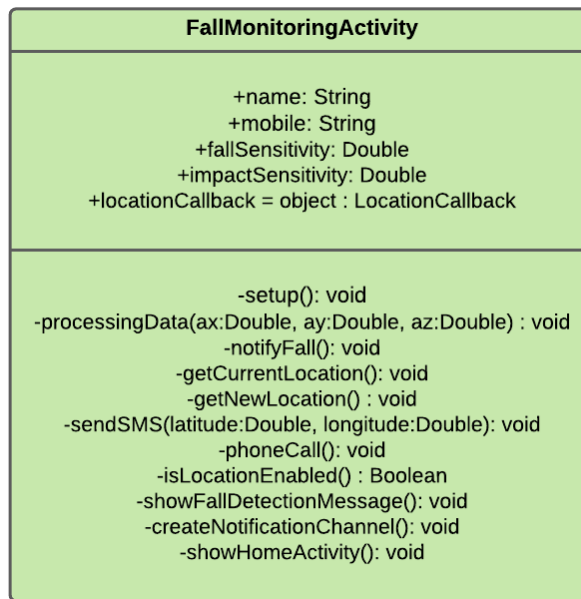


Figura 3.13. Diagrama de la clase *FallMonitoringActivity*

3.3. Estructura de la base de datos

La aplicación hace uso de dos de los productos ofrecidos por Firebase para el almacenamiento de información. El primer producto es *Firebase Realtime Database*, se trata de una base de datos en tiempo real que emplea datos con formato JSON estructurados en árbol como muestra la Figura 3.14. En esta base de datos se almacena toda la información referente al usuario, como el nombre de usuario y contraseña, la información del perfil de usuario y la configuración de la aplicación. Esto permite que, al volver a iniciar sesión en el mismo dispositivo o en otro, esta información se recupere y no sea necesario volver a introducirla.

El segundo producto que se va a utilizar es *Firebase Storage*. Su principal funcionalidad es el almacenamiento de ficheros, puesto que permite crear carpetas para su agrupación. Este sistema se utiliza para almacenar y ordenar los ficheros CSV creados con los datos sobre la actividad. La Figura 3.15 muestra cómo se estructura en carpetas el servidor dedicándose a cada una de las posibles actividades-tipo a ejecutar un monitorizar.

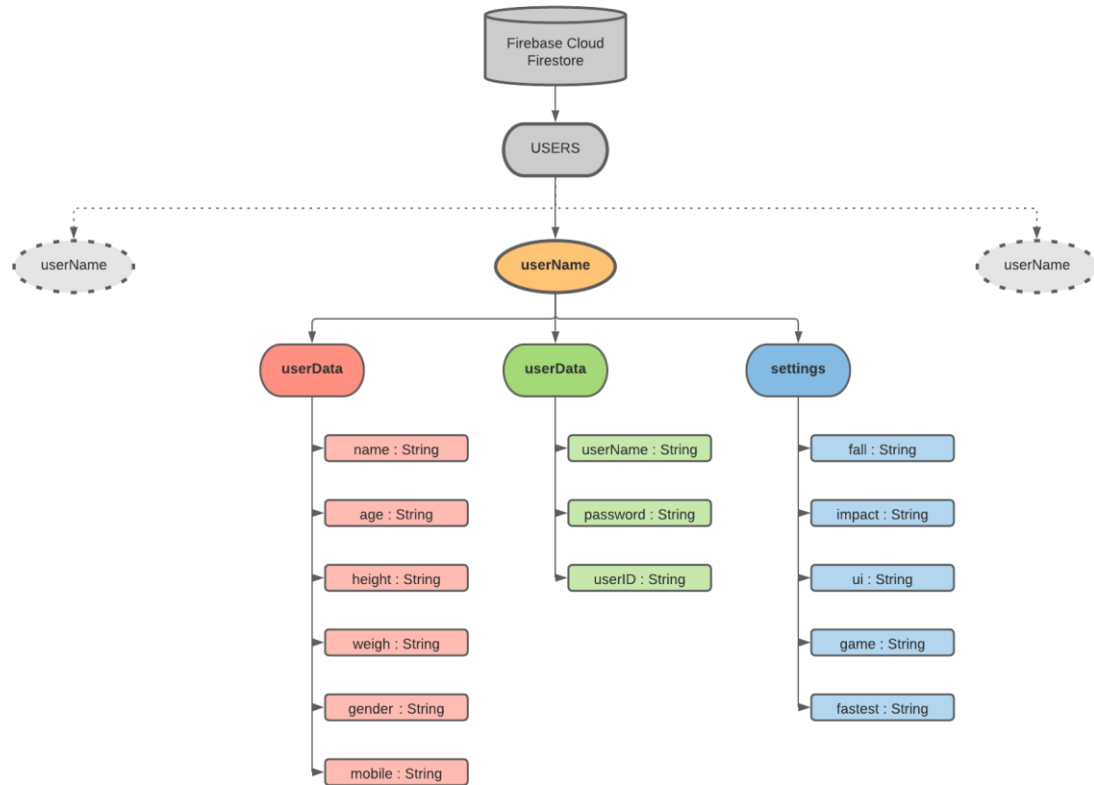


Figura 3.14. Estructura de los datos en *Firebase Realtime Database*

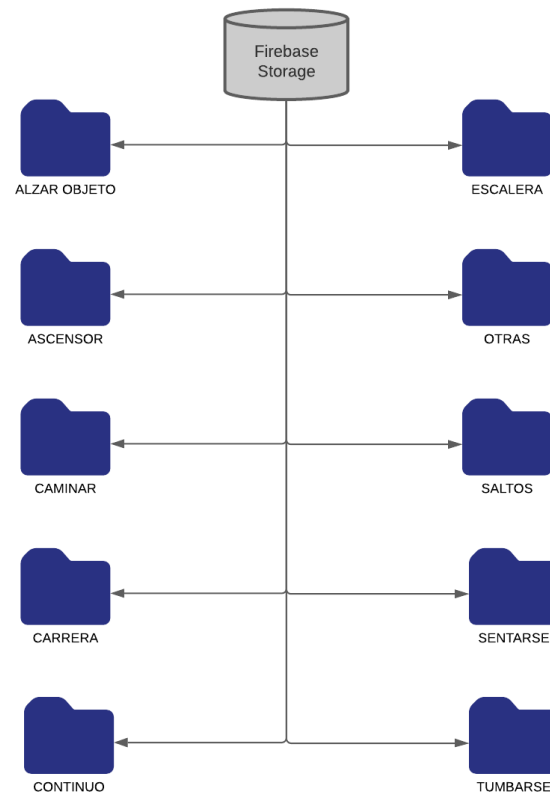


Figura 3.15. Estructura de archivos en *Firebase Storage*

Capítulo 4. Verificación y pruebas

Este capítulo busca proporcionar información sobre la validez del software desarrollado a través de la realización de una serie de pruebas basadas principalmente en la comprobación de los requisitos establecidos en el segundo capítulo.

4.1. Sistema de pruebas

En el campo de las pruebas de software existen diversos métodos que permiten encontrar defectos y evaluar su calidad. En el entorno de pruebas que se va a definir, se establecerán dos grupos de verificación basados en la accesibilidad a los elementos evaluables. Los métodos escogidos se conocen como “pruebas de caja blanca” y “pruebas de caja negra” [18].

Una prueba de caja blanca asume el total acceso y conocimiento del código fuente, siendo el entorno de pruebas software el IDE (*Integrated Development Environment*) de desarrollo empleado (Android Studio) y el entorno hardware un *smartphone* conectado a un PC mediante un puerto USB, haciendo uso de la herramienta ADB (*Android Debug Bridge*) para la depuración.

Una prueba de caja negra busca validar la funcionalidad, siendo únicamente necesario establecer el entorno hardware, el *smartphone*, ya que el entorno software será la propia aplicación. Para verificar que los archivos se cargan correctamente en la base de datos y que los parámetros de configuración se actualizan en tiempo real se utilizará la plataforma web de Firebase.

Una vez establecidos los tipos de pruebas a realizar, se pasa a describir las principales características de los entornos utilizados, pues los resultados podrían diferir en equipos diferentes.

En primer lugar, el smartphone utilizado es un dispositivo *Samsung Galaxy A31* con sistema operativo Android cuyas principales características se muestran en la Tabla 4.1.

SAMSUNG GALAXY A31	
PANTALLA	Super AMOLED de 6,4 pulgadas Resolución FullHD (2.400 x 1.080 píxeles)
PROCESADOR	Octa Core (Dual 2 GHz + Hexa 1,7 Ghz)
MEMORIA RAM	4 GB
ALMACENAMIENTO INTERNO	64 GB + microSD
CÁMARA TRASERA	48 MP f/2.0 Gran angular 8 MP f/2.2 Profundidad 5 MP f/2.4 Macro 5 MP f/2.4
CÁMARA DELANTERA	20 MP f/2.2
BATERÍA	5.000 mAh con carga rápida 15W
SISTEMA OPERATIVO	Android 10 con One UI 2.0
CONECTIVIDAD	4G/LTE, WiFi ac, Bluetooth, NFC, USB-C, jack 3,5 mm
OTROS	Lector de huellas bajo la pantalla
DIMENSIONES Y PESO	73,1 x 159,3 x 8,6 mm 185 g

Tabla 4.1. Características del *Samsung Galaxy A31* [19]

Este dispositivo permite emplear dos mecanismos de comunicación para aquellas pruebas que requieren conexión a Internet.

- *LTE (Long Term Evolution)*: es un estándar para comunicaciones inalámbricas de transmisión de datos para dispositivos móviles y

terminales de datos. Opera en la banda de frecuencias comprendida entre 1.8 GHz y 2.6 GHz. En nuestro entorno concreto de pruebas, el operador que ofrece este servicio es Movistar.

- Wi-Fi: es una red inalámbrica de área local basada en el estándar IEEE 802.11. El proveedor del servicio de conexión a Internet es igualmente Movistar.

El PC empleado como entorno hardware es un ordenador de sobremesa construido a medida. Posee un procesador AMD FX de ocho núcleos a 3.61 GHz, 16 GB de memoria RAM DDR3 y un disco duro de 1 TB que ejecuta el sistema operativo Windows 10 Home de 64 bits. El entorno software, por su parte, es la versión 2020.30.1 de Android Studio y la versión 31 de Android SDK.

Para las pruebas de monitorización de actividades, el dispositivo se situará en la cintura usando una riñonera deportiva, como la mostrada en la Figura 4.1, que permita acercar el dispositivo al cuerpo.



Figura 4.1. Elemento de sujeción del dispositivo

Las pruebas referentes a la detección de caídas se centrarán en dejar caer el dispositivo a una superficie blanda desde una altura aproximada a la de la cintura. Con esto se busca simular una caída sin poner en riesgo al usuario ni al dispositivo.

4.2. Pruebas realizadas

En este apartado se procederá a elaborar una tabla que recoja las pruebas que se realizarán en función a los requisitos ya establecidos en el segundo capítulo.

En caso de que sean favorables, mostraran el cumplimiento de los requisitos de funcionalidades del sistema.

Nº	ID del requisito	Descripción de la prueba	Resultado
1	3	Crear un nuevo usuario para verificar la información requerida por el sistema. Resultado: al pulsar en “CREAR USUARIO” abre una actividad que permite el registro de nuevos usuarios, comprobando si el nombre de usuario introducido ya está en uso.	Favorable
2	4	Añadir en el proceso de creación de usuario la información sobre nombre, edad, peso, altura, género y teléfono de contacto. Resultado: la creación de usuario permite crear un perfil completo del usuario que posteriormente puede ser modificado.	Favorable
3	7	Hacer uso de la configuración para modificar los datos introducidos. Resultado: tras pulsar “CONFIGURACIÓN” se accede a una actividad que permite modificar la información introducida por el usuario, guardándose tras pulsar en “GUARDAR Y SALIR”.	Favorable
4	5	Comprobar la actualización de los datos en tiempo real en la consola de la plataforma Firebase. Resultado: tras las pruebas anteriores, una vez pulsado el botón “GUARDAR”, se observa como en la consola de Firebase se actualiza la información del usuario en el servidor.	Favorable
5	6	Iniciar la aplicación sin haber autenticado ningún usuario. Resultado: la aplicación se inicia en una actividad que permite introducir el nombre de usuario y la contraseña para autenticarse en la aplicación.	Favorable
6	13	Tratar de acceder a la monitorización de caídas haciendo uso de un perfil de usuario incompleto. Resultado: el sistema comprueba si dispone del nombre y el teléfono de contacto. En caso afirmativo inicia la actividad, en caso contrario muestra un mensaje y cambia el color del botón de configuración a rojo.	Favorable
7	23	Intentar acceder a la monitorización de actividad mediante un perfil de usuario incompleto. Resultado: el sistema comprueba si dispone del nombre y el teléfono de contacto. En caso afirmativo inicia la actividad, en caso contrario muestra un mensaje y cambia el color del botón de configuración a rojo.	Favorable
8	14	Dejar caer el dispositivo con el sonido activado. Resultado: cuando se detecta una posible caída, el sistema lanza una notificación sonora audible.	Favorable

Tabla 4.2. Tabla de las pruebas realizadas

Nº	ID del requisito	Descripción de la prueba	Resultado
9	15	<p>Simular una caída para comprobar que el aviso se cancela correctamente.</p> <p>Resultado: tras activarse aparece un mensaje de aviso con dos botones y una cuenta regresiva. Tras pulsar en cancelar, el aviso queda anulado con éxito.</p>	Favorable
10	16	<p>Simular dos caídas para comprobar que tanto el aviso manual como el aviso automático tras finalizar la cuenta atrás se ejecutan correctamente.</p> <p>Resultado: ambas formas de notificar al contacto de emergencia funcionan correctamente al establecerse una llamada con el contacto de emergencia.</p>	Favorable
11	17	<p>Simular una caída y activar el aviso de emergencia para comprobar que el altavoz de la llamada esta activo.</p> <p>Resultado: mediante la propia llamada y al observar las opciones activas en el dispositivo se verifica que el altavoz está activo.</p>	Favorable
12	19	<p>Simular la caída con el módulo GPS activo.</p> <p>Resultado: el mensaje SMS recibido contiene la URL con la geolocalización, constatándose su exactitud mediante <i>Google Maps</i>.</p>	Favorable
13	20	<p>Simular la caída con el GPS desactivado.</p> <p>Resultado: el mensaje SMS recibido solo avisa del suceso, pero carece de la ubicación al discriminar si el GPS está activo o no.</p>	Favorable
14	21	<p>Iniciar la monitorización de caídas y tratar de volver a la actividad anterior mediante el botón de la interfaz y el botón físico del dispositivo.</p> <p>Resultado: la acción de volver atrás queda invalidada mientras la monitorización está activa, ya que la app deshabilita el botón de la interfaz y muestra un mensaje que indica que es necesario detener el proceso para el botón físico.</p>	Favorable
15	30	<p>Iniciar la monitorización de actividad e intentar volver a la activada anterior mediante el botón de la interfaz y el botón físico del dispositivo.</p> <p>Resultado: se obtienen los mismos resultados que con el experimento anterior.</p>	Favorable
16	29	<p>Realizar la monitorización de cualquier actividad para comprobar que, una vez se ha detenido el proceso, el archivo se guarda y almacena en el servidor.</p> <p>Resultado: desde la consola de Firebase se comprueba que el fichero de la actividad se guarda correctamente en la carpeta asignada a dicha actividad.</p>	Favorable

Tabla 4.2. Tabla de las pruebas realizadas (Continuación)

Nº	ID del requisito	Descripción de la prueba	Resultado
17	32	Realizar tanto la monitorización de caídas como la monitorización de actividad con la pantalla apagada para comprobar que el sistema responde correctamente. Resultado: la detección de caídas responde de la misma forma que con la pantalla encendida, al igual que el registro de actividad, comparando los ficheros CSV.	Favorable
18	-	Prueba de caja blanca: se realizarán las pruebas anteriores haciendo uso de la herramienta ADB (Android Debug Bridge) de Android Studio que permite realizar la depuración del código conectando el dispositivo al PC mediante el cable USB. Resultado esperado: tras las diversas acciones sobre la aplicación, se pueden observar en la ventana de depuración los correspondientes mensajes.	Favorable

Tabla 4.2. Tabla de las pruebas realizadas (Continuación)

Capítulo 5. Base de datos de movimientos

En este capítulo se describirá la base de datos creada a partir de los experimentos de recogida de datos mediante la función de monitorización de actividad.

5.1. Organización de la base de datos

La base de datos se divide en una serie de carpetas, correspondiendo cada una de ellas a las actividades que se pueden registrar con la aplicación y que el sujeto experimental debe elegir en la app justo antes de ejecutar el correspondiente movimiento. Las carpetas y su contenido se muestran a continuación:

- “ALZAR OBJETO”: la actividad registrada consiste en levantar un objeto del suelo varias veces.
- “ASCENSOR”: esta carpeta contiene las mediciones de un usuario que sube y baja en ascensor.
- “CAMINAR”: los datos almacenados en esta carpeta deben corresponderse con una caminata de algunos minutos de duración.
- “CARRERA”: esta información se corresponde con los datos recogidos de una carrera ligera.
- “CONTINUA”: esta carpeta contiene archivos con las medidas de actividades de la vida cotidiana registradas durante largos periodos de tiempo (es decir, cuando el usuario tiene libertad de movimiento y no realiza ninguna actividad prefijada)
- “ESCALERAS”: los datos sobre actividad que se almacenan en esta carpeta se obtienen subiendo y bajando tramos de escalera.

- “OTROS”: es la carpeta que almacena todas las actividades que el usuario nombra mediante etiquetas descriptivas.
- “SALTOS”: esta actividad recoge las medidas que se obtiene cuando el usuario da varios saltos consecutivos.
- “SENTARSE” la actividad registrada consiste en sentarse y levantarse de un asiento varias veces.
- “TUMBARSE”: esta carpeta contiene los datos del usuario tras tumbarse y levantarse de una cama varias veces seguidas.

5.2. Experimento de captura de movimientos

Este apartado recoge en la Tabla 5.1 un resumen con la información de los experimentos realizados, tanto la información anonimizada de los participantes, como las actividades que han realizado. Es importante mencionar que cada voluntario que ha participado en el experimento ha realizado las actividades marcadas una única vez.

Sujeto	Edad	Altura (cm)	Peso (Kg)	Género	Alzar Objeto	Ascensor	Caminar	Carrera	Escaleras	Salto	Sentarse	Tumbarse
01	29	190	107	M	✓	✓	✓	✓	✓	✓	✓	✓
02	28	185	87	M	✓	✓	✓	✓	✓	✓	✓	✓
03	22	170	72	M	✓	x	✓	✓	✓	✓	✓	x
04	24	183	63	M	✓	x	✓	✓	✓	✓	✓	✓
05	27	178	73	M	✓	x	✓	✓	✓	✓	✓	✓
06	27	157	45	F	✓	x	✓	✓	✓	✓	✓	✓
07	31	164	63	F	✓	x	✓	✓	✓	✓	✓	✓
08	29	178	69	M	✓	x	✓	✓	✓	✓	✓	✓
09	47	156	59	F	✓	x	✓	✓	✓	✓	✓	✓
10	46	190	84	M	✓	x	✓	✓	✓	✓	✓	✓
11	25	156	46	F	✓	x	✓	✓	✓	✓	✓	x
12	21	162	54	F	✓	x	✓	✓	✓	✓	✓	x
13	29	188	75	M	✓	x	✓	✓	✓	✓	✓	✓

Tabla 5.1. Tabla de las actividades realizadas. M: Masculino, F: Femenino

El servidor de Firebase es privado, teniendo acceso únicamente el propietario de la cuenta, por lo que los ficheros han sido descargados y compartidos en un servidor de *Google Drive* accesible mediante el siguiente enlace:

https://drive.google.com/drive/folders/1enuwu8Yh2wtby5h6sBMRImRmXy6_8tPd?usp=sharing

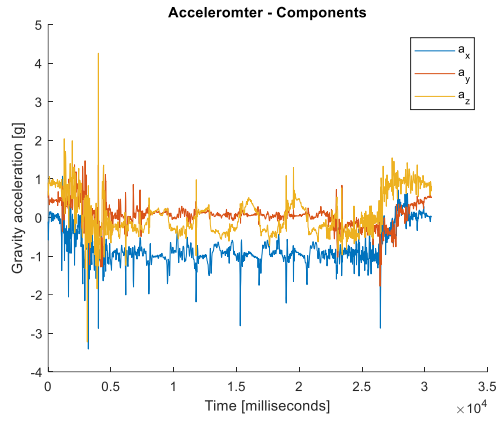
Cada fichero creado para almacenar las medidas del acelerómetro y del giróscopo presentan la estructura de la Figura 5.1. Comienzan por una cabecera que ocupa siete líneas, mostrando primero el identificador del usuario, seguido de su información personal. A continuación, está la fecha de inicio y el tipo de actividad, terminando con la tasa de muestreo configurada, información del dispositivo y las unidades en las que se miden las diferentes magnitudes.

Las medidas de los sensores se encuentran en filas distintas, ya que Android no permite tomas muestras simultaneas de varios sensores. Cada muestra tiene asignada una marca de tiempo, con precisión de milisegundos, que ocupa la primera columna, identificada en la Figura 5.1 como columna “A”. Las columnas “B”, “C” y “D” contienen las muestras del acelerómetro, medidas en g, terminado con las muestras del giróscopo, medidas en rad/s, que ocupan las columnas “E”, “F”, y “G”.

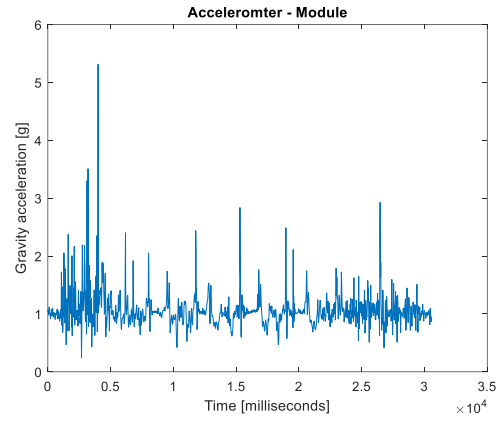
	A	B	C	D	E	F	G
1	# User ID sujeto01						
2	# Peso: 107 -Altura: 190 -Edad: 29 -Genero: Masculino						
3	# Fecha y hora del comienzo de la prueba 27/10/2021 10:13:55						
4	# Tipo de actividad "Caminar"						
5	# Tipo de muestreo: SENSOR_DELAY_GAME						
6	# Características del dispositivo. Version ANDROID: 30 -Modelo: SM-A315G -Fabricante: samsung						
7	# Acelerómetro [g] ---Giróscopo [rad/s]						
8	TimeStamp	ax	ay	az	gx	gy	gz
9	10:13:55.298	0.0600387	0.3458327	0.7548768			
10	10:13:55.301				-3.0659328	0.4263839	1.5430456
11	10:13:55.316	-0.2230706	0.281889	0.9567142			
12	10:13:55.319				-3.0659328	0.4263839	1.5430456
13	10:13:55.336	-0.4373551	0.3475411	1.21737			
14	10:13:55.339				-3.0659328	0.4263839	1.5430456
15	10:13:55.356	-0.3243554	0.2125761	1.4077366			
16	10:13:55.359				-3.0659328	0.4263839	1.5430456
17	10:13:55.376	-0.1256908	0.1737705	1.1744155			
18	10:13:55.379				-0.6181957	0.444099	0.1582141
19	10:13:55.396	0.0344124	0.065408	0.8546973			
20	10:13:55.399				-0.3475823	0.6939429	0.3915646
21	10:13:55.416	-0.0883496	0.2472325	0.9547618			
22	10:13:55.420				-0.8539896	0.8374963	0.3304781
23	10:13:55.436	-0.1745027	0.1684012	1.0421351			
24	10:13:55.440				-1.323745	0.580322	0.3115413
25	10:13:55.456	-0.1291076	0.0131792	1.1121803			
26	10:13:55.459				-1.2003502	0.7819075	0.4025602
27	10:13:55.476	-0.2679776	0.032704	1.172463			
28	10:13:55.479				-0.7232645	0.2962697	0.1875356
29	10:13:55.496	-0.2169691	0.1198333	1.0726426			

Figura 5.1. Estructura de la base de datos.

Para terminar este capítulo, en la Figura 5.2 y la Figura 5.3 se representan gráficas con las componentes y el módulo de las medidas de ambos sensores para la actividad “SENTARSE”, realizada por el “sujeto01”.

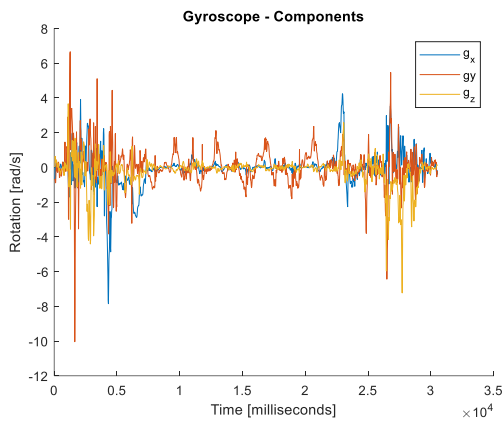


a)

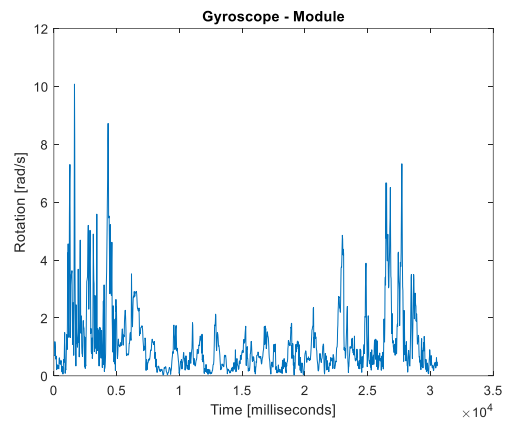


b)

Figura 5.2. Medidas del acelerómetro para la actividad "SENTARSE"



a)



b)

Figura 5.3. Medidas del giróscopo para la actividad "SENTARSE"

Capítulo 6. Manual de instalación y uso

Este capítulo describe el proceso de instalación de la aplicación en el dispositivo Android, además de las especificaciones mínimas requeridas para dicha operación. Asimismo, se incluye un manual de uso de la herramienta que permita al usuario sacar el máximo rendimiento a las funcionalidades implementadas.

6.1. Manual de instalación

Para poder instalar de forma satisfactoria la aplicación en el *smartphone* Android, es necesario que el dispositivo cumpla una serie de características mínimas.

- Sistema operativo: Android 6.0 (*Marshmallow*).
- Memoria RAM: 1 GB.
- Espacio libre en la memoria interna: 500 MB.
- Sensores disponibles: acelerómetro y giróscopo.

Una vez se posee un dispositivo con las características mencionadas, se inicia el proceso de instalación de la *app*. Para ello se requiere de un fichero con extensión APK (*Android aPplication pacKage*) que se genera mediante el entorno de desarrollo utilizado tras un proceso de compilación y empaquetado del proyecto.

Para comenzar con la instalación, es necesario transferir al almacenamiento interno del dispositivo el archivo APK de la aplicación, localizado en el directorio raíz del DVD adjunto al proyecto con el nombre “MovNFall.apk”.

Al tratarse de una aplicación que no ha sido descargada a través de *Google Play Store*, será necesario cambiar la configuración de seguridad para permitir la

instalación de aplicaciones de origen desconocido, tal y como muestra la Figura 6.1. este paso es obligatorio pues, al desconocerse el origen de la app, el sistema operativo requiere del permiso del usuario para verificar que es consciente del riesgo de este tipo de instalaciones, ya que podrían provenir de fuentes no fiables y contener software malicioso. La forma de proceder para otorgar los permisos para instalar una aplicación de origen desconocido puede diferir de un modelo de móvil a otro.

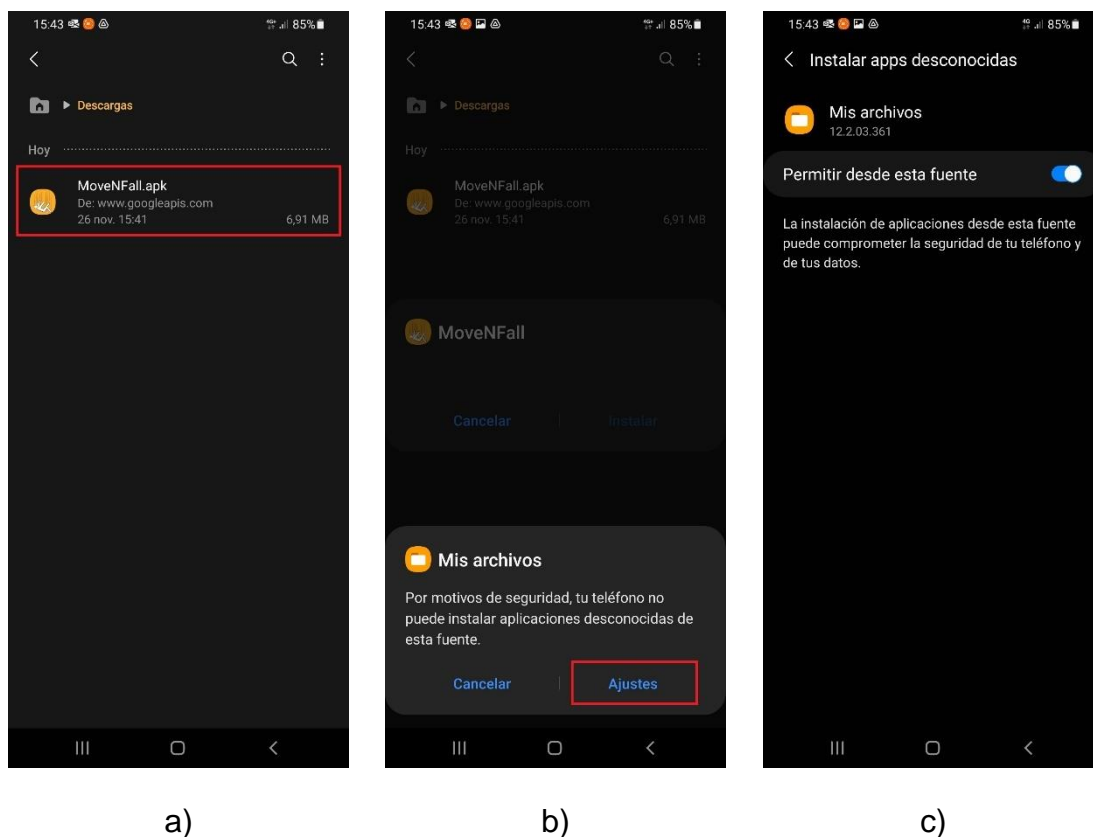


Figura 6.1. pasos para habilitar la instalacion de aplicaciones de origen desconocido

Una vez habilitado el permiso, basta con buscar el archivo APK en el directorio al que haya sido transferido y tras pulsar sobre el mismo se inicia el proceso de instalación (véase la Figura 6.2). Como se ve en la Figura 6.2 b), es posible que, pese a haber permitido la instalación de aplicaciones con origen desconocido, sea necesario indicar que el usuario está seguro de querer continuar con el proceso.

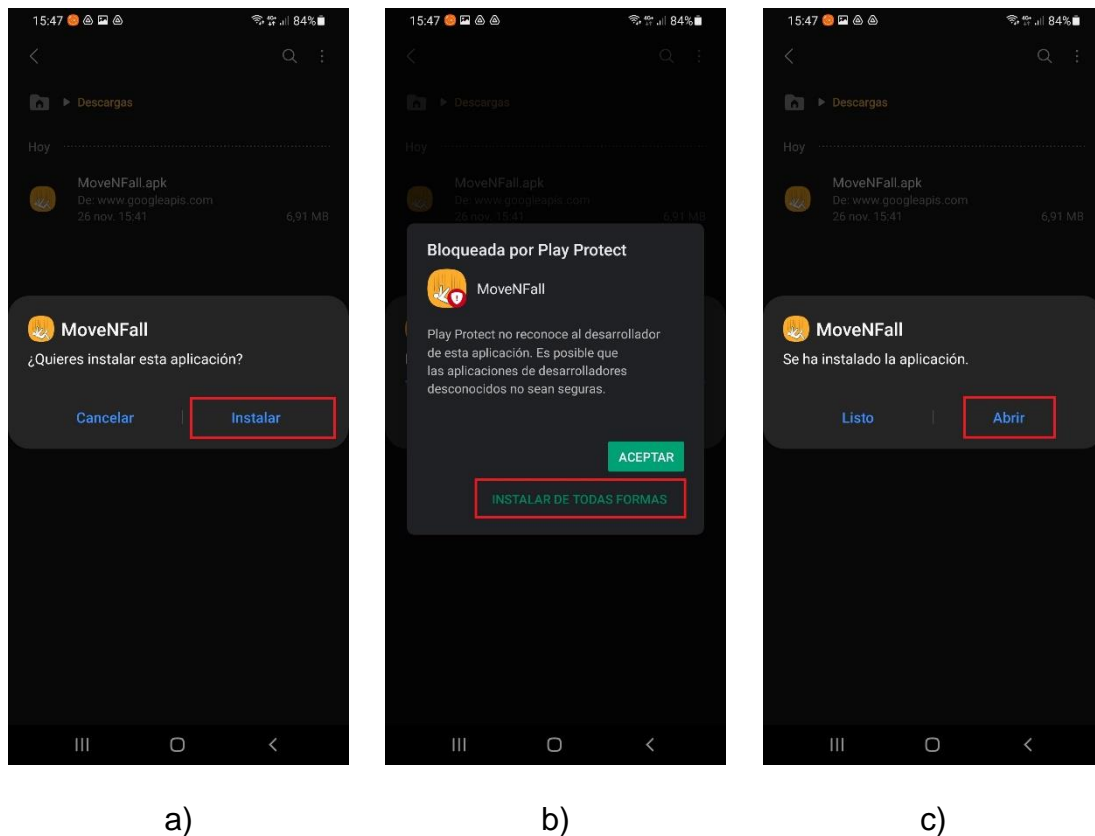


Figura 6.2. Pasos para instalar la aplicación

Aunque la aplicación solicita los permisos necesarios para la utilización de los diferentes módulos del dispositivo al ejecutarse por primera vez, es posible otorgarlos tras el proceso de instalación, desde el menú de ajustes del dispositivo, como se muestra en la Figura 6.3.

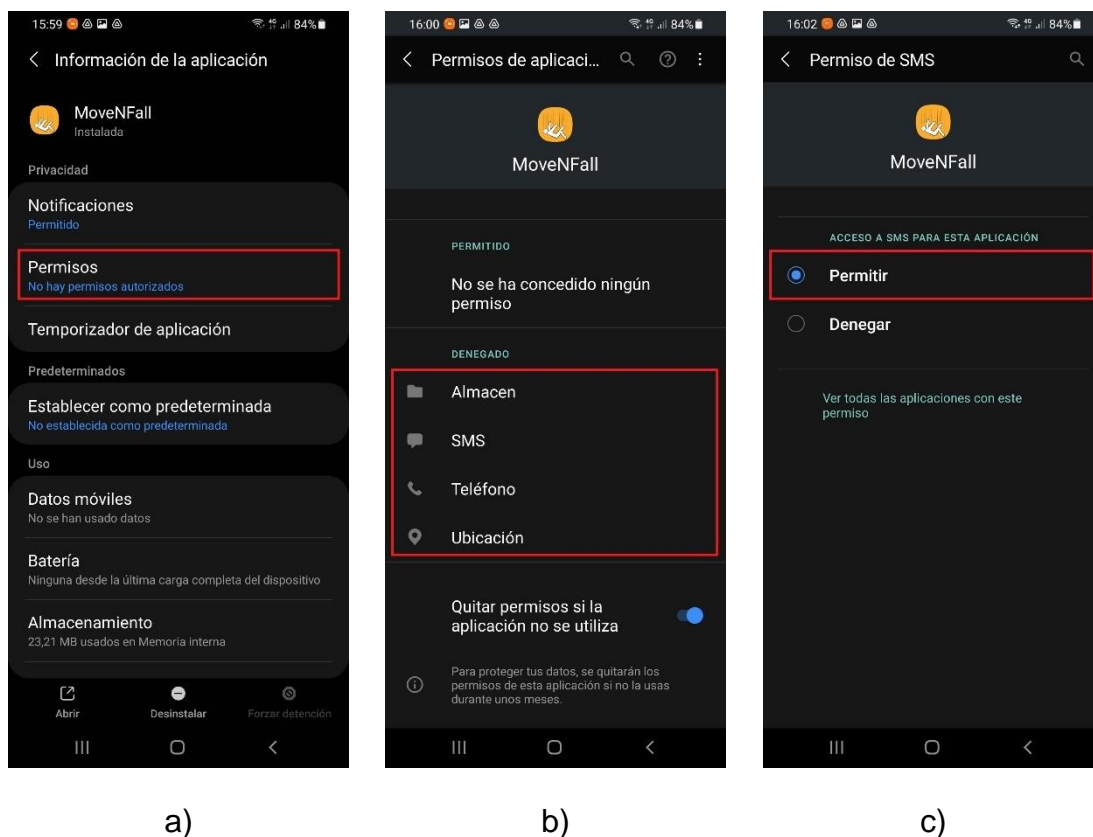


Figura 6.3. Habilitación de permisos desde la configuración del dispositivo

Si todos estos pasos se han seguido correctamente, la aplicación estará instalada y lista para funcionar.

6.2. Manual de uso

En esta sección se presenta un manual de uso de la aplicación en el que se explica el funcionamiento completo del software, desde la primera ejecución hasta el cierre de sesión.

6.2.1. Primera ejecución de la aplicación

Cuando la aplicación se ejecuta por primera vez, si no se otorgaron los permisos desde el menú de configuración del *smartphone*, el sistema solicitará los permisos necesarios al usuario automáticamente, como muestra la Figura 6.4. Los permisos necesarios para el correcto funcionamiento de la aplicación son:

- Almacenamiento: permite a la aplicación acceder al almacenamiento interno del dispositivo para poder crear los ficheros CSV y almacenar información obtenida de los sensores.
- SMS: otorga a la aplicación permisos para poder enviar mensajes SMS.
- Teléfono: da la capacidad al sistema de iniciar una llamada telefónica a través de la aplicación nativa del teléfono.
- Ubicación: autoriza a la aplicación para obtener la ubicación del usuario con el objetivo de incluir la geolocalización en el SMS enviado al contacto de emergencia.

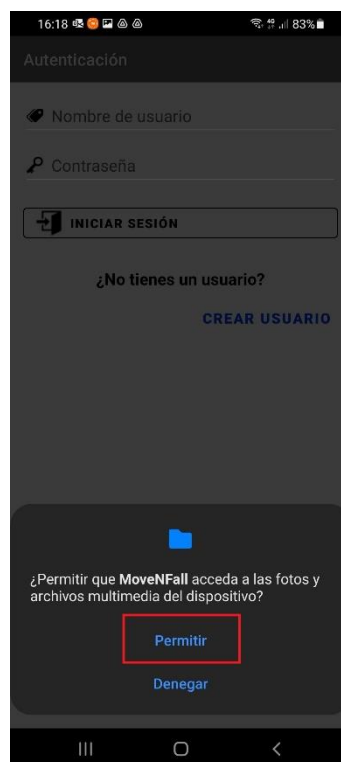


Figura 6.4. Habilitación de permisos desde la aplicación

Si se dispone de un usuario y una contraseña, estos podrán introducirse en la sección indicada en la Figura 6.5 a), usando el botón “INICIAR SESIÓN” para acceder a la aplicación. Por el contrario, si no se dispone de un usuario, podrá crearse pulsando en “CREAR USUARIO” en el apartado marcado de la Figura 6.5 b).



Figura 6.5. Menú de autenticación de la herramienta

6.2.2. Creación de un nuevo usuario

La Figura 6.6 a) muestra la ventana de creación de usuario. Los campos mínimos requeridos para crear uno son “Nombre de usuario”, “Contraseña” y “Repetir contraseña”. La Figura 6.6 b) muestra a modo de ejemplo, estos campos cumplimentados. Tras pulsar en “REGISTRAR USUARIO”, en el caso de que los campos estén vacíos o contengan errores como caracteres no válidos o contraseñas diferentes, se mostrará un mensaje de error y se marcará en rojo el campo correspondiente como indica la Figura 6.6 c).

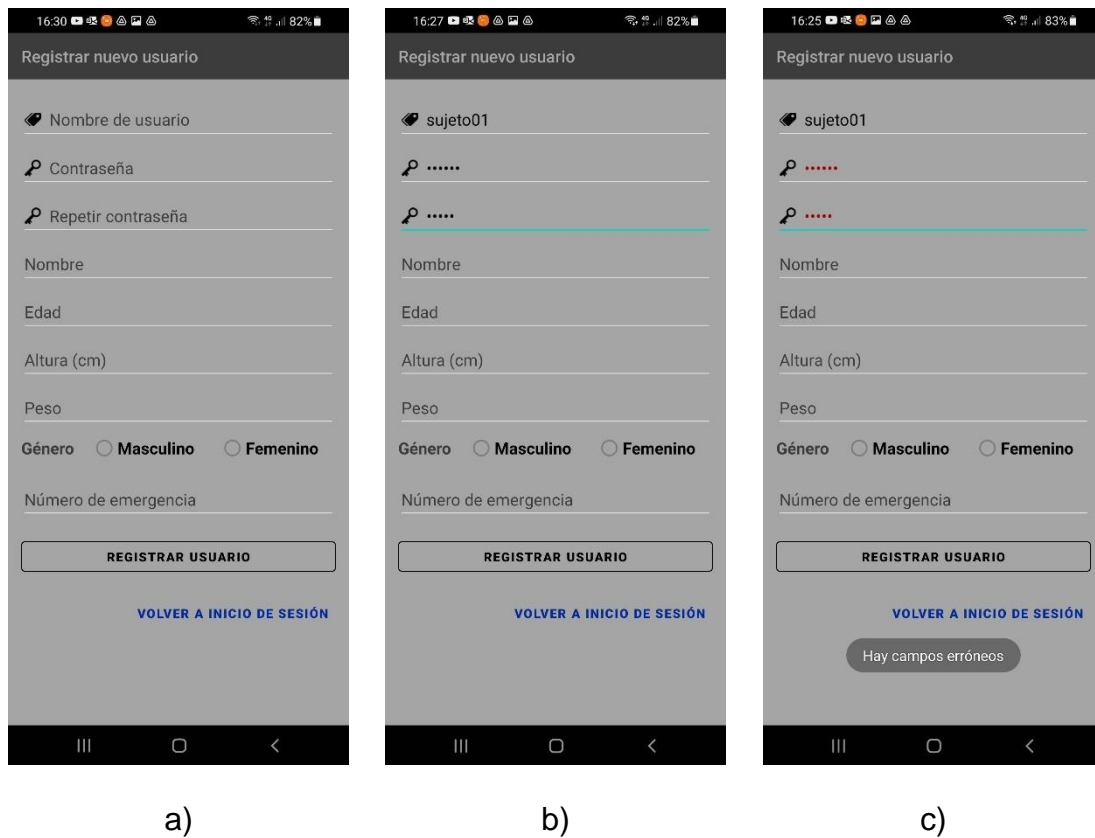


Figura 6.6. Ventana de creación de usuario incompleto

Los campos “Nombre”, “Edad”, “Altura”, “Peso”, “Género” y “Número de emergencia” pueden añadirse en este punto opcionalmente, tal y como muestra la Figura 6.7.

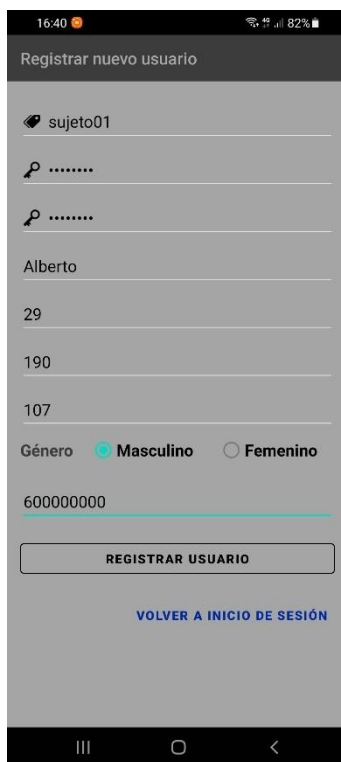


Figura 6.7. Ventana de creación de usuario completo

Pulsando en “REGISTRAR USUARIO” se completa el proceso de creación y alta del usuario tras el cual la aplicación regresa al menú principal. En el caso de haber pulsado en “VOLVER A INICIO DE SESIÓN”, el programa volverá a la pantalla de autenticación.

6.2.3. Inicio de sesión con un usuario creado

Como se mencionó en el apartado 6.2.1, un usuario que ya dispone de una cuenta, puede iniciar sesión en la ventana de autenticación, introduciendo su nombre de usuario y contraseña en los campos correspondientes en la ventana de autenticación, como muestra Figura 6.8. Tras pulsar en “INICIAR SESIÓN” se autentica al usuario, pasando a la pantalla del menú principal.

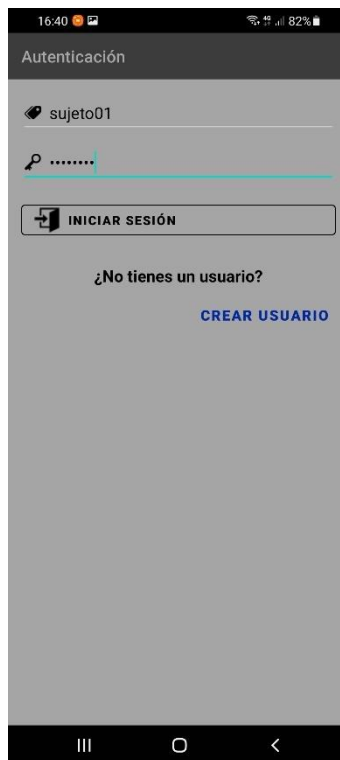


Figura 6.8. Autenticación de un usuario

6.2.4. Menú principal

En la Figura 6.9 se representa el menú principal de la aplicación, que permite acceder mediante botones a las funciones principales del software: monitorización de actividad y monitorización de caídas, así como acceder a la configuración del sistema o cerrar sesión.

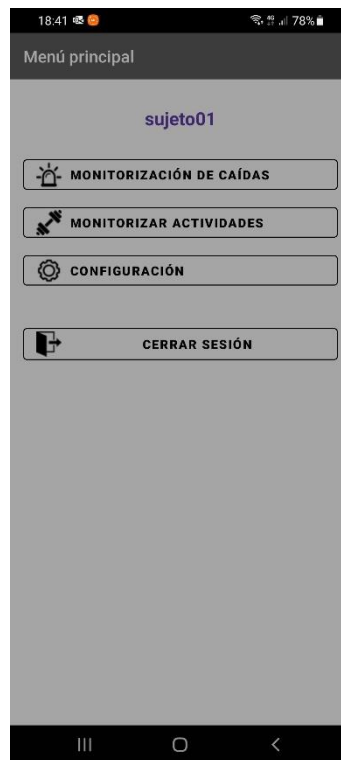


Figura 6.9. Menú principal de la herramienta

Una vez que el usuario está autenticado, cada vez que abre la aplicación accede directamente a esta ventana, hasta que cierra la sesión mediante el botón correspondiente.

Para acceder a las funciones de monitorización, la aplicación se asegura de que el perfil del usuario está totalmente cumplimentado, mostrando un mensaje de error y marcando en rojo el botón de configuración como muestran la Figura 6.10 a) y la Figura 6.10 b).

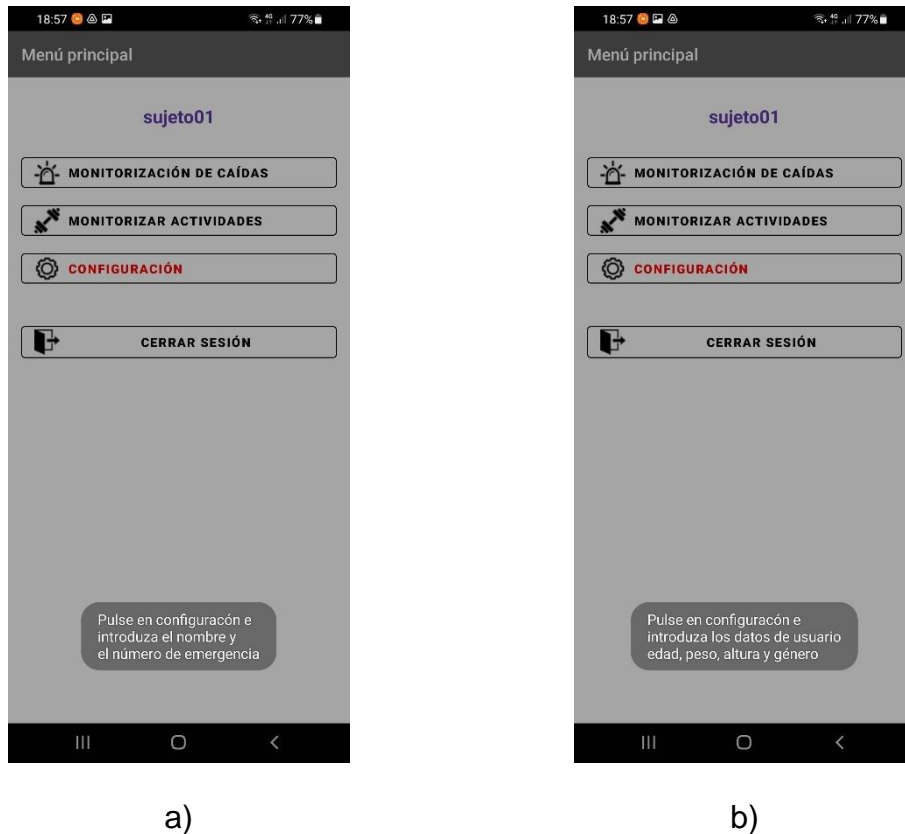


Figura 6.10. Acceso a las funcionalidades con un usuario incompleto

6.2.5. Configuración de la aplicación

La ventana de configuración de la *app* que se muestra en la Figura 6.11, se divide en dos secciones bien diferenciadas. Por un lado, están los datos del usuario, correspondientes la misma información que la ya mencionada en el apartado 6.2.2, permitiendo completar el perfil o actualizar la información, Figura 6.11 a).

Por otro lado, está la opción de configuración del sistema, (véase la Figura 6.11 b). esta posibilita seleccionar un valor para el umbral que determina la fase de caída libre y el umbral de la fuerza del impacto que permite interpretar los datos como una caída. En esta sección también está presente la selección de la velocidad de muestreo, mediante botones que funcionan de forma excluyente, dado que la activación de uno desactiva a los demás, pudiendo seleccionarse solo una opción.



Figura 6.11. Ventana de configuración

El elemento final es el botón “GUARDAR Y SALIR”, cuya función es guardar la nueva configuración en el dispositivo y en el servidor, y regresar finalmente al menú principal.

6.2.6. Monitorización de actividad

El proceso de monitorización de actividad comienza en la ventana de selección, a la que se accede tras pulsar el botón “MONITORIZAR ACTIVIDADES” del menú principal. En este nuevo menú se puede elegir entre diversas actividades a realizar (como se muestra en la Figura 6.12 a). A continuación, se presenta una breve explicación de las diferentes actividades que debería ejecutar el usuario:

- Escaleras: subir y bajar algunos tramos de escalera.
- Tumbarse: tumbarse y levantarse varias veces de un sofá o una cama.
- Caminar: caminar durante algún tiempo.
- Carrera: correr a un ritmo ligero durante cierto tiempo.

- Saltos: dar algunos saltos.
- Sentarse: sentarse y levantarse de una silla o sillón algunas veces.
- Alzar objetos: levantar varias veces un objeto del suelo.
- Monitorización continua: monitorización de la actividad genérica del usuario durante un largo periodo de tiempo. El objetivo de esta opción es permitir el seguimiento del usuario durante su vida diaria, sin aplicar etiquetas de actividades concretas.
- Otras: realización de actividades personalizadas por el usuario.

Cuando se pulsa en cualquiera de los botones, se da paso a la ventana de monitorización, Figura 6.12 b). Existe un caso particular, que se da cuando se pulsa en el botón “OTRAS”. Esta acción abre una ventana similar, pero con un cuadro de texto que permite al usuario nombrar la actividad que va a realizar, Figura 6.12 c). esta opción debería utilizarse si el usuario piensa ejecutar movimientos no recogidos en las actividades ofrecidas en el menú.

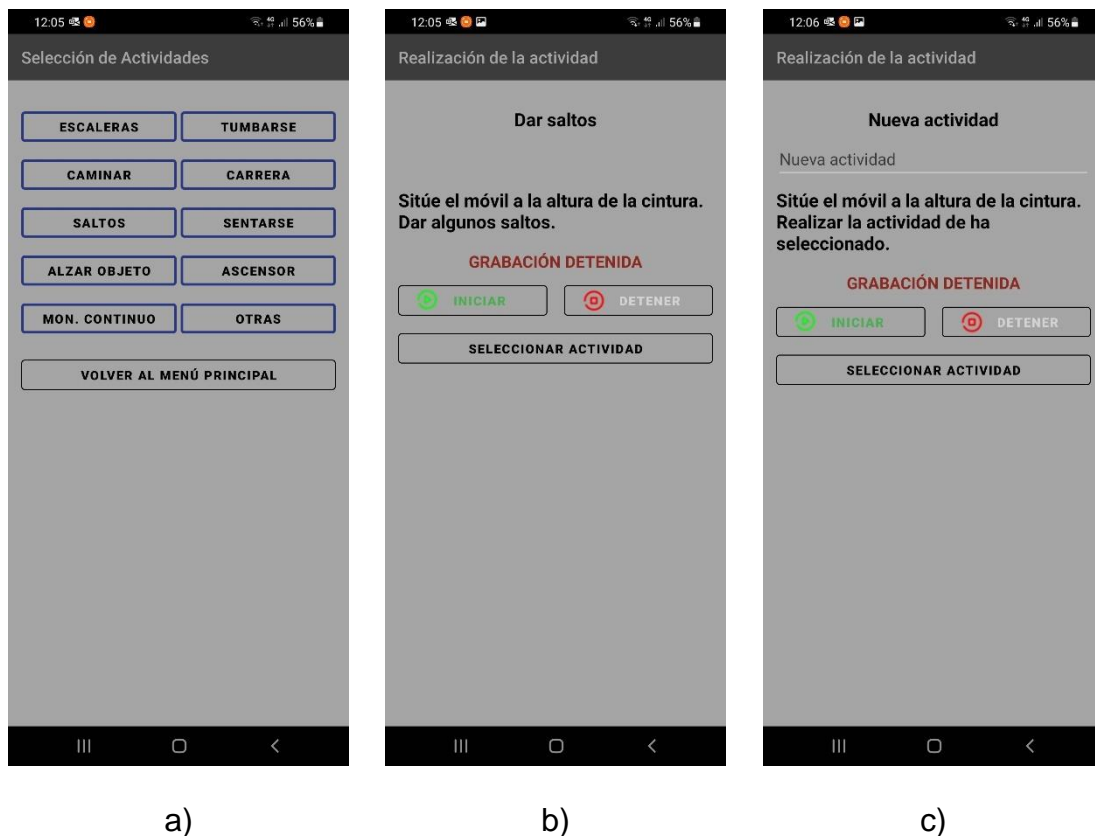


Figura 6.12. Selección y realización de una actividad

La grabación de movimiento comienza tras pulsar en “INICIAR”, mostrando un mensaje y cambiando el texto de “GRABACIÓN DETENIDA” a “GRABANDO ACTIVIDAD” como muestra la Figura 6.13 a).

Para detener el proceso, basta con pulsar en el botón de “DETENER”, proceso que inicia la carga del archivo en el servidor mediante una ventana emergente de progreso, (véase la Figura 6.13 b). La funcional de monitorización concluye con un mensaje que indica que el proceso se ha llevado a cabo con éxito, (véase Figura 6.13 c).

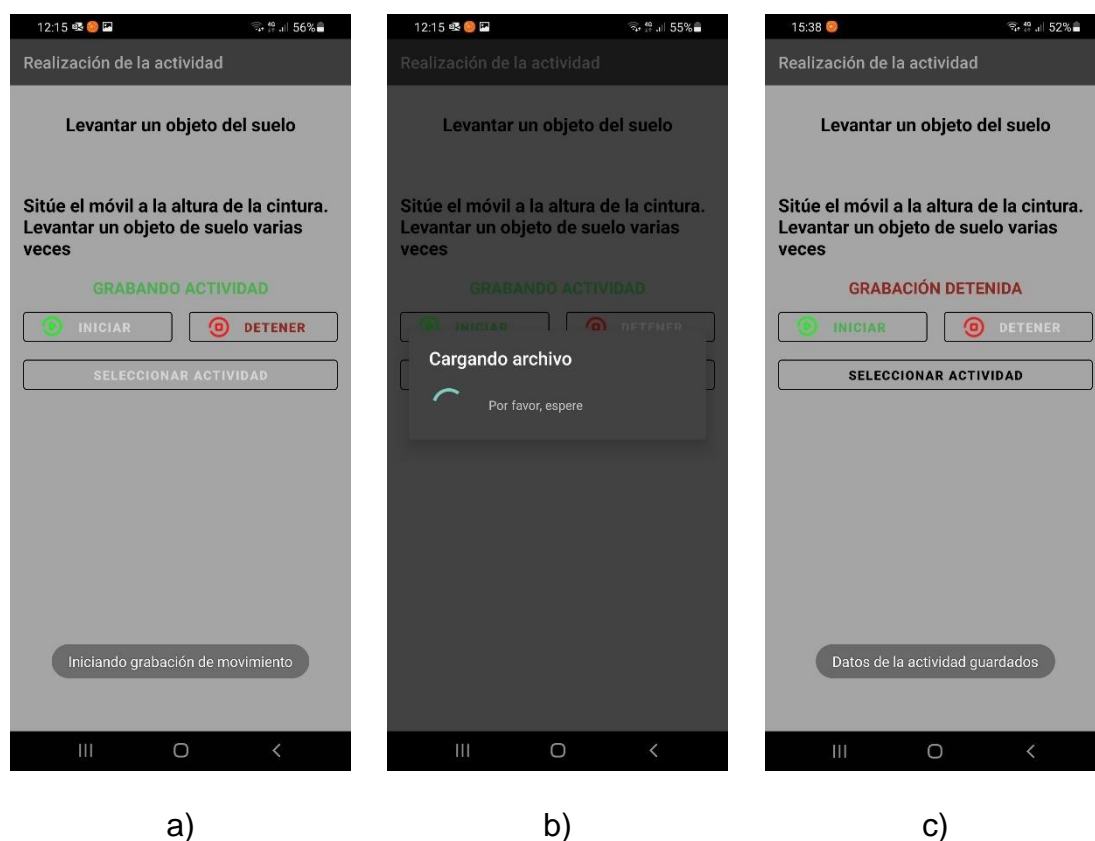


Figura 6.13. Proceso de monitorización de actividad

Una vez transcurridos unos segundos tras la finalización de la monitorización de actividad, la aplicación vuelve a la ventana de selección de actividad.

Un último aspecto a considerar sobre la función de monitorización de actividad es que el sistema verifica que el usuario ha introducido un nombre para la actividad, mostrando un mensaje y marcando el cuadro de texto en rojo en el caso de que se haya etiquetado el ejercicio (tal y como se ilustra en la Figura 6.14).

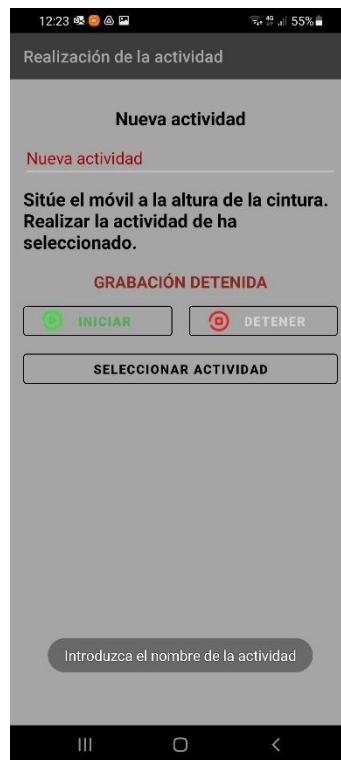
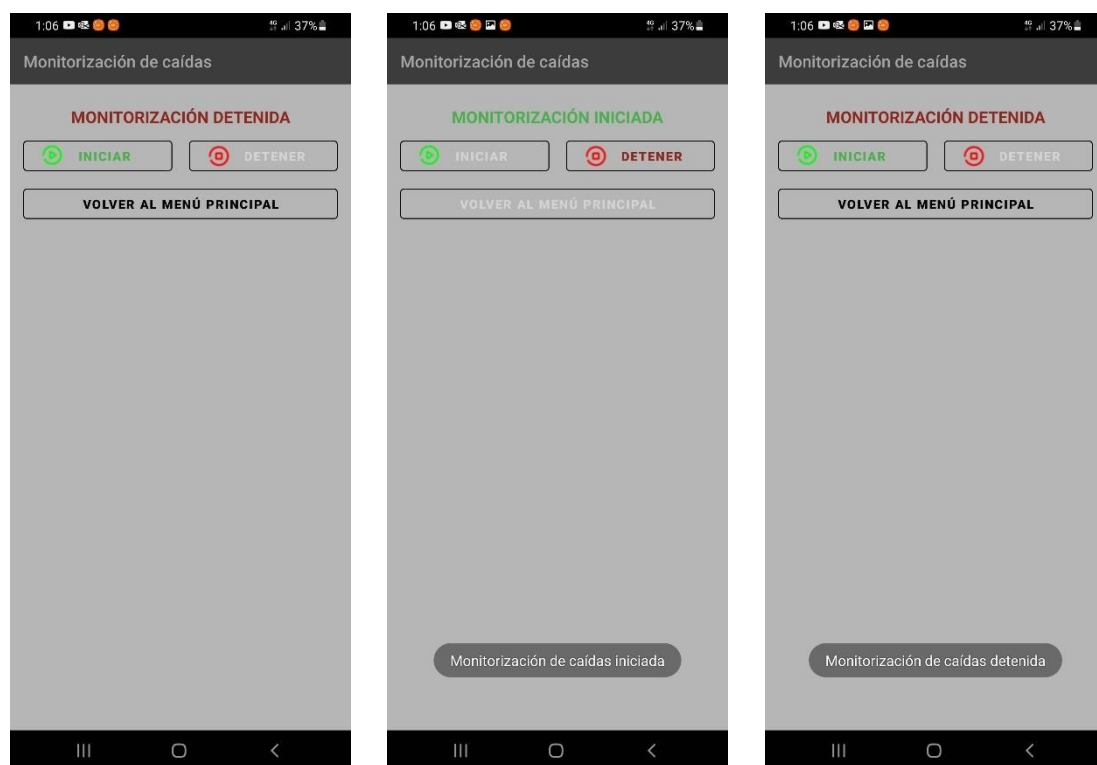


Figura 6.14. Actividad “OTRAS” sin etiquetar

6.2.7. Monitorización de caídas

A esta funcionalidad se accede tras pulsar en el botón “MONITORIZACIÓN DE CAÍDAS” del menú principal, accediendo a la ventana donde se realiza el seguimiento correspondiente (como muestra la Figura 6.15 a).

El proceso de detección de caídas se inicia mediante el botón “INICIO”, tras el que se visualiza un mensaje que avisa al usuario de que la monitorización ha comenzado y cambiando el mensaje de “MONITORIZACIÓN DETENIDA” a “MONITORIZACIÓN INICIADA” como muestra la Figura 6.15 b). Para detener la detección, basta con pulsar en “DETENER”, momento en que se pausa el seguimiento y se muestran avisos similares a los indicados previamente, (véase la Figura 6.15 c).



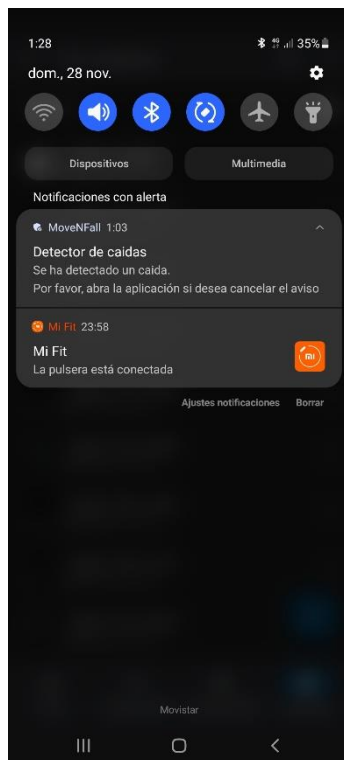
a)

b)

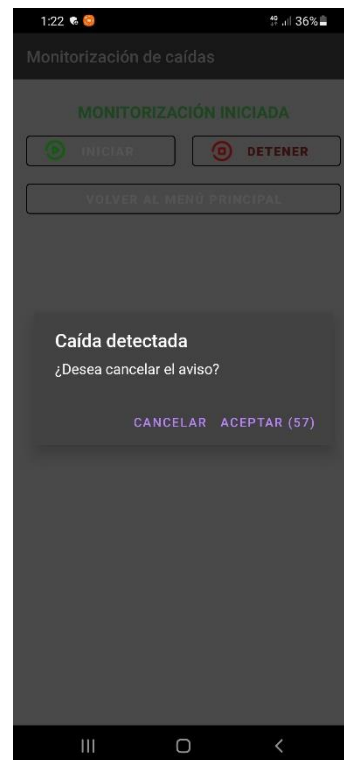
c)

Figura 6.15. Ventana de la funcionalidad de detección de caídas

Cuando se detecta la caída, la aplicación muestra un aviso de posible caída a través de la barra de notificaciones, Figura 6.16 a), y un mensaje emergente en la aplicación, Figura 6.16 b). Este mensaje emergente permite cancelar el aviso al contacto de emergencia o dar la alarma de forma manual. Cuando la cuenta atrás de un minuto expira sin cancelar el aviso, la alarma se dispara de forma automática.



a)



b)

Figura 6.16. Detección de una posible caída

El protocolo de aviso consiste en una llamada al contacto de emergencia manteniendo el altavoz encendido como se muestra en la Figura 6.17 a) y un SMS al teléfono indicado durante el alta del usuario en el sistema. Si la ubicación del dispositivo está activada, el mensaje contendrá la ubicación del usuario (Figura 6.17 b). En caso contrario, el mensaje solo dará el aviso del accidente al contacto de emergencia (Figura 6.17 c).

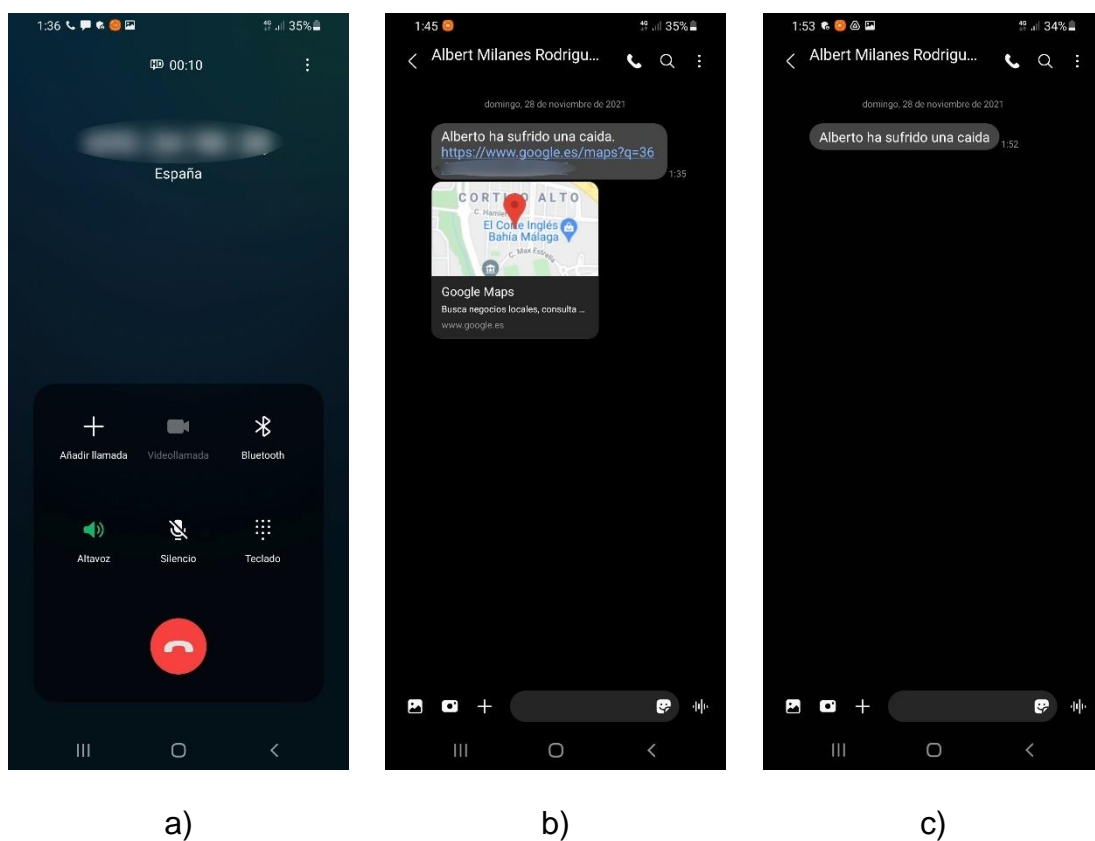


Figura 6.17. Tipos de avisos al contaacto de emergencia

6.2.8. Cierre de sesión del usuario.

Esta última sección del manual de uso trata sobre el cierre de sesión. Este proceso se ejecuta tras pulsar, en la pantalla principal, en el botón de cierre de sesión, como muestra la Figura 6.18.

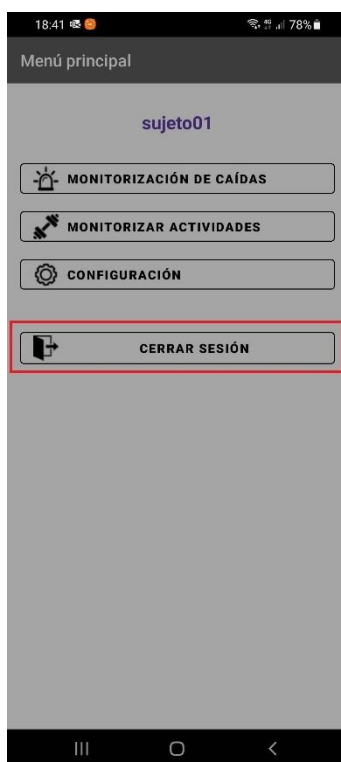


Figura 6.18. Cierre de sesión en la aplicación

El cierre de sesión implica que la aplicación elimina la información del usuario del dispositivo, terminando por volver a la ventana de autenticación, de la que se comenzó hablando en los primeros apartados del manual.

Capítulo 7. Conclusiones y trabajo futuro

7.1. Tareas realizadas

Tras un análisis del problema inicial y haber establecido una serie de objetivos con los que alcanzar una solución, se han llevado a cabo una serie de tareas que se resumen en los siguientes puntos:

- Se ha realizado un estudio de mercado en busca de soluciones similares para analizar su coste frente a las características ofrecidas, con el fin de desarrollar una solución de bajo coste para un dispositivo habitual en la vida diaria.
- Con el objetivo de desarrollar una aplicación Android, se han estudiado los recursos que ofrece el lenguaje Kotlin orientado a objetos.
- Con el fin de obtener información con la que poder mejorar los algoritmos de reconocimiento de actividad, se ha desarrollado un sistema de captura de movimiento basado en los sensores de un dispositivo *smartphone*.
- Se ha desarrollado un algoritmo sencillo de detección de caídas (basado en umbrales) con la capacidad de alertar a un contacto de emergencia a través de una llamada telefónica y un mensaje SMS.
- Se ha establecido un entorno de pruebas con las que el usuario puede escoger entre una serie de actividades predefinidas o crear las suyas propias con el objetivo de crear bases de datos para el desarrollo y valoración de sistemas de monitorización vestibles.

- Se han adquirido los conocimientos para manejar una base de datos en tiempo real.
- Se han analizado diferentes medios para elaborar una batería de pruebas con las que garantizar el correcto funcionamiento de la aplicación.

7.2. Discusión y conclusiones

El principal objetivo de este proyecto era encontrar una solución tecnológica con la que tratar de solucionar un problema social muy frecuente en la actualidad como es poder socorrer a personas de edad avanzada imposibilitadas por una caída.

Uno de los principales problemas para poder encontrar una solución es desarrollar un sistema que sea capaz de detectar una caída con una alta probabilidad de acierto. En cuanto a los sistemas de detección de caídas comerciales, uno de los principales contras que se han encontrado es que los usuarios deben recordar llevar el dispositivo encima y pegado al cuerpo, por lo que, al escoger el *smartphone*, un objeto habitual en la vida moderna, reduce este problema.

Por otra parte, el haber escogido un algoritmo basado en umbrales origina un sistema válido para la detección de caídas, pero creemos que ofrecería menor precisión que sistemas más complejos basados en el análisis del movimiento de las personas con algoritmos más sofisticados. Esto se debe a que, según como el usuario personalice los umbrales, puede ser que las caídas suaves no sean detectadas o que los movimientos comunes de mayor intensidad sean confundidos con una caída.

Se puede concluir que es un sistema funcional que sirve para detectar caídas, pero lo que resulta realmente útil es la función de registro de actividad, pues adquiere información con la que desarrollar sistemas de mayor precisión.

Como objetivos secundarios al proyecto, me he iniciado en el aprendizaje de un nuevo lenguaje de programación, Kotlin, además de familiarizarme con el entorno de desarrollo Android Studio junto al uso de ciertas funcionalidades para el manejo de usuarios y bases de datos como puede ser la herramienta Firebase de Google.

7.3. Líneas futuras

El trabajo realizado presenta una serie de recursos con las que poder generar nuevas vías de desarrollo. En este apartado se presentan algunas de ellas que pueden resultar interesantes.

Una primera línea de desarrollo podría ser establecer un sistema de evaluación sistemática del software desarrollado para detectar posibles errores de funcionamiento debidos a situaciones no previstas durante la etapa de desarrollo, pues usuarios no relacionados con el proyecto podrían llegar a situaciones que no fueron contempladas en un primer momento.

Otra línea interesante sería desarrollar una aplicación que discrimine entre usuario y contacto de emergencia, de modo que puedan asociarse las cuentas para usar la aplicación para dar el aviso de emergencia, añadiendo una nueva forma de comunicación.

Continuando en la línea anterior, se propone usar las funciones de geolocalización para crear un protocolo de rastreo, útil para personas que padecen enfermedades similares a la demencia, de modo que se pueda establecer una zona segura, para que, cuando salga de ella, se active una función de rastreo para conocer su ubicación en tiempo real.

Finalmente, con el fin de mejorar la precisión del detector de caídas, usar sistemas de inteligencia artificial que analicen el movimiento del usuario y sean capaces de discernir si se ha producido una caída en base a las medidas de los sensores, mejorando el actual algoritmo basado en umbrales.

Apéndice A. Presupuesto de elaboración

En este apéndice se incluye un listado de los costes de desarrollo del sistema. El presupuesto los siguientes conceptos:

- Coste de amortización de equipos: *smartphone* y PC.
- Coste del trabajo en horas del análisis y desarrollo de la aplicación.
- Presupuesto total.

La relación de costes no contempla el IVA, los gastos generales ni el beneficio industrial. Al haberse empleado Android Studio, que es una licencia de software libre, y Matlab con una versión de prueba, no se contempla el coste de las respectivas licencias.

Componente	Coste unitario (€)	Periodo de amortización	Periodo de uso	Coste total (€)
PC a medida	900 €	36 meses	3 meses	75.00 €
<i>Smartphone Samsung Galaxy A31</i>	269 €	36 meses	3 meses	22.42 €
Total				97.42 €

Tabla en Apéndice A.1. Amortización de equipos

Actividad	Horas
Coordinación del TFG	5
Análisis de requisitos	20
Desarrollo hardware/software del prototipo	160
Realización de las pruebas	10
Redacción de la memoria	120
Preparación de la presentación	15
Horas totales	320
Coste por hora 30 €	Coste total de la mano de obra 9600 €

Tabla en Apéndice A.2. Horas invertidas en la realización del proyecto (mano de obra)

Concepto	Coste (€)
Coste de componentes y material fungible	97.42 €
Mano de obra	9600 €
Coste total	9697.42 €

Tabla en Apéndice A.3. Presupuesto total

Apéndice B. Estructura del DVD

El DVD adjunto a la memoria del proyecto contiene los siguientes ficheros:

- **Software.zip:** se trata de un archivo comprimido que incluye el código del proyecto MoveNFall.
- **Memoria.zip:** archivo comprimido que incluye tanto la memoria en formato pdf como los diagramas de clases y flujogramas utilizados en la memoria.
- **Instalador.zip:** archivo comprimido que contiene el archivo MoveNFall.apk, cuyo proceso de instalación se explica paso a paso en el sexto capítulo del presente documento.

Referencias

- [1] Indicadores de estructura de la población. Proporción de personas mayores de cierta edad por provincia.
<https://www.ine.es/jaxiT3/Datos.htm?t=1488#!tabs-mapa> [Último acceso el 2 de diciembre del 2021]
- [2] Cifras de población. Indicadores demográficos básicos. Datos porvisionales a 1 de enero de 2021.
https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176951&menu=ultiDatos&idp=1254735572981 [Último acceso el 2 de diciembre del 2021]
- [3] Proyección de la esperanza de vida al nacimiento por periodo. Brecha de género. España.
https://www.ine.es/jaxi/Datos.htm?path=/t00/mujeres_hombres/tablas_1/I0/&file=d01003.px#!tabs-grafico [Último acceso el 2 de diciembre del 2021]
- [4] Número de hogares unipersonales por comunidad y ciudades autónomas según sexo, edad y estado civil.
<https://www.ine.es/jaxi/Tabla.htm?path=/t20/p274/serie/prov/p02/I0/&file=02014.px&L=0> [Último acceso el 2 de diciembre del 2021]
- [5] Revista Española de Geriatria y Gerontología. Caídas en la población anciana española: incidencia, consecuencias y factores de riesgo.
<https://www.elsevier.es/es-revista-revista-espanola-geriatria-gerontologia-124-pdf-S0211139X15000931> [Último acceso el 2 de diciembre del 2021]
- [6] Tracmi. Reloj con llamadas SOS y localizador GPS.
<https://tracmi.es/tienda/relojes-gps/reloj-con-llamadas-sos-y-localizador-gps/> [Último acceso el 2 de diciembre del 2021]

Referencias

- [7] Tracmi. Reloj GPS S4 Mayores. <https://tracmi.es/tienda/relojes-gps/anticaida-s4-mayores-suscripcion-mensual/> [Último acceso el 2 de diciembre del 2021]
- [8] Neki. Detector de caídas GPS Nock Senior. <https://neki.es/colgante-detector-caidas-localizador-gps-ancianos.html> [Último acceso el 2 de diciembre del 2021]
- [9] PANION. M-GUARD PRO. <https://www.panion.es/produkt/m-guard-pro/> [Último acceso el 2 de diciembre del 2021]
- [10] Apple. Detección de caídas con *Apple Watch*. <https://support.apple.com/es-es/HT208944> [Último acceso el 2 de diciembre del 2021]
- [11] *Mobile Operating System Market Share Worldwide*. <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-202110> [Último acceso el 2 de diciembre del 2021]
- [12] *Distribution data for Android*. <https://androiddistribution.io/#/> [Último acceso el 2 de diciembre del 2021]
- [13] Documentación de Firebase para Android. https://firebase.google.com/docs/android/setup?authuser=0#next_steps [Último acceso el 2 de diciembre del 2021]
- [14] Web oficial de Eclipse. <https://www.eclipse.org/> [Último acceso el 2 de diciembre del 2021]
- [15] Web oficial de Android Studio. <https://developer.android.com/studio> [Último acceso el 2 de diciembre del 2021]
- [16] Descripción general de proyectos. <https://developer.android.com/studio/projects?hl=es-419> [Último acceso el 2 de diciembre del 2021]
- [17] Arquitectura de la plataforma. <https://developer.android.com/guide/platform?hl=es-419> [Último acceso el 2 de diciembre del 2021]
- [18] Pruebas de caja blanca y caja negra. <https://www.testernoderno.com/caja-blanca-vs-caja-negra/> [Último acceso el 2 de diciembre del 2021]

Referencias

- [19] Análisis y características del *Samsung Galaxy A31*.
<https://www.xatakamovil.com/analisis/samsung-galaxy-a31-analisis-caracteristicas-precio-especificaciones> [Último acceso el 2 de diciembre del 2021]